

This Page Is Inserted by IFW Operations
and is not a part of the Official Record

BEST AVAILABLE IMAGES

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images may include (but are not limited to):

- BLACK BORDERS
- TEXT CUT OFF AT TOP, BOTTOM OR SIDES
- FADED TEXT
- ILLEGIBLE TEXT
- SKEWED/SLANTED IMAGES
- COLORED PHOTOS
- BLACK OR VERY BLACK AND WHITE DARK PHOTOS
- GRAY SCALE DOCUMENTS

IMAGES ARE BEST AVAILABLE COPY.

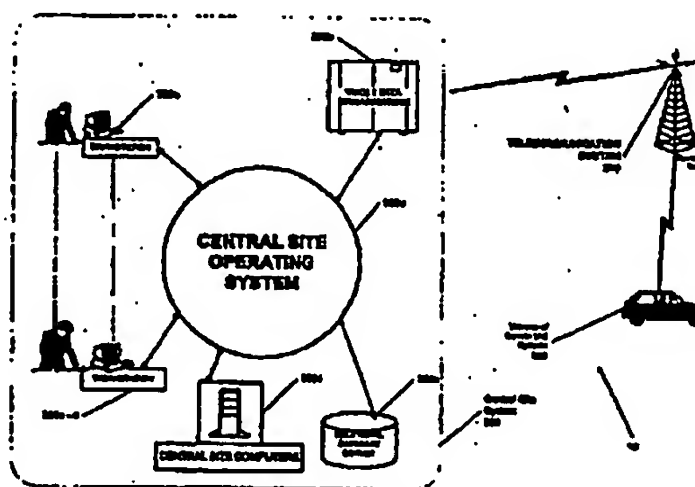
**As rescanning documents *will not* correct images,
please do not report the images to the
Image Problem Mailbox.**



INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(51) International Patent Classification 6 : H04Q 7/00, 7/22, 7/20	A1	(11) International Publication Number: WO 98/10602
		(43) International Publication Date: 12 March 1998 (12.03.98)
(21) International Application Number: PCT/US97/15362		(81) Designated States: CA, JP, European patent (AT, BE, CH, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE).
(22) International Filing Date: 29 August 1997 (29.08.97)		
(30) Priority Data: 60/025,578 6 September 1996 (06.09.96) US		
(71) Applicant: MOBILE SECURITY COMMUNICATIONS, INC. [US/US]; Suite 500, 2856 Amwiler Road, Atlanta, GA 30360 (US).		
(72) Inventor: MARLOWE, John, M.; 1058 Highway 332, Pendergrass, GA 30567 (US).		
(74) Agents: FERRELL, John, S. et al.; Carr & Ferrell, Suite 200, 2225 East Bayshore Road, Palo Alto, CA 94303 (US).		Published With international search report. Before the expiration of the time limit for amending the claims and to be republished in the event of the receipt of amendments.

(54) Title: MOBILE SECURITY SYSTEM



(57) Abstract

A mobile security system providing an object-oriented operating system (300a) that enables a non-homogeneous universe of remote mobile units (100) to communicate with a central site system (300). The operating system defined by a plurality of interactive application programs consisting of a communication server, a resource manager, a monitor application, a database server (300c) and an administrator. The communication server comprises a plurality of objects that connect the remote unit to the central site system. The communication server interactively communicates to the resource manager a need for the monitor application to be interactively connected to the communication server. The resource manager comprises a plurality of objects that control the operating systems' resources. The resource manager receiving the request from the communication server facilitates the connection to the next appropriate operator of a workstation (300e-n) containing the monitor program. The monitor application comprises a plurality of objects that command and control the interaction between the mobile remote unit and the central site system.

FOR THE PURPOSES OF INFORMATION ONLY

Codes used to identify States party to the PCT on the front pages of pamphlets publishing international applications under the PCT.

AL	Albania	ES	Spain	LS	Lesotho	SI	Slovenia
AM	Armenia	FI	Finland	LT	Lithuania	SK	Slovakia
AT	Austria	FR	France	LU	Luxembourg	SN	Senegal
AU	Australia	GA	Gabon	LV	Latvia	SZ	Swaziland
AZ	Azerbaijan	GB	United Kingdom	MC	Monaco	TD	Chad
BA	Bosnia and Herzegovina	GE	Georgia	MD	Republic of Moldova	TG	Togo
BB	Barbados	GH	Ghana	MG	Madagascar	TJ	Tajikistan
BE	Belgium	GN	Guinea	MK	The former Yugoslav Republic of Macedonia	TM	Turkmenistan
BF	Burkina Faso	GR	Greece	ML	Mali	TR	Turkey
BG	Bulgaria	HU	Hungary	MN	Mongolia	TT	Trinidad and Tobago
BJ	Benin	IE	Ireland	MR	Mauritania	UA	Ukraine
BR	Brazil	IL	Israel	MW	Malawi	UG	Uganda
BY	Belarus	IS	Iceland	MX	Mexico	US	United States of America
CA	Canada	IT	Italy	NE	Niger	UZ	Uzbekistan
CF	Central African Republic	JP	Japan	NL	Netherlands	VN	Viet Nam
CG	Congo	KE	Kenya	NO	Norway	YU	Yugoslavia
CH	Switzerland	KG	Kyrgyzstan	NZ	New Zealand	ZW	Zimbabwe
CI	Côte d'Ivoire	KP	Democratic People's Republic of Korea	PL	Poland		
CM	Cameroon	KR	Republic of Korea	PT	Portugal		
CN	China	KZ	Kazakhstan	RO	Romania		
CU	Cuba	LC	Saint Lucia	RU	Russian Federation		
CZ	Czech Republic	LI	Liechtenstein	SD	Sudan		
DE	Germany	LK	Sri Lanka	SE	Sweden		
DK	Denmark	LR	Liberia	SG	Singapore		
EE	Estonia						

MOBILE SECURITY SYSTEM

Background of the Invention

1. Field Of The Invention

5 The invention relates generally to a mobile security system containing a remote unit and a central site monitoring system. In particular the invention is directed to an operating system for the central site monitoring system. The invention is more particularly directed to an object-oriented operating system that provides a flexible architecture. That architecture allows interactive communication
10 between the remote unit and the central site system.

2. Description of Related Art

 Personal and vehicle security has become of paramount importance in the United States and in foreign countries. Current statistics suggest that vehicle theft
15 cost the consumer approximately \$10.5 billion. These costs are attributed to actual replacement cost of a stolen vehicle, law enforcement efforts to address vehicle thefts and related crimes, and personal expenditures that relate directly to the crime of vehicle theft.

20 Mobile security systems of the past have made vehicular theft more difficult for amateur thieves but have made little impact on the professional vehicle thief. The violence associated with vehicle theft by amateur thieves has increased and the professional thief has become more sophisticated. An indicator of the increased violence is the rise in the number of vehicles stolen at gun point —
25 carjacking. The professional thief, knowing the limitations of present day vehicle security, will simply by-pass alarm devices and steal the vehicle. The professional thief feels confident that even if an alarm is sounded by a vehicle the public will most likely ignore the alarm because false alarms are prevalent in present day

vehicle alarm systems. The professional thief also knows that communication between the vehicle, if it exists, and a central monitoring station is limited to monitoring. The central station may receive a communique from the vehicle but it may not be able to discern the nature or reason for the communique. An example of this would be the vehicle alarm is activated but the central station does not have means to tell the difference between a false alarm and an in-process vehicle theft.

Present day central stations have operating systems that allow passive monitoring of a vehicle that contains an associated alarm system. The operating system may include some type of vehicle positioning and tracking, displayed on a device such as the monitor of a computer. Vehicle tracking provides limited information to the operating system if there is no central station communication with the vehicle.

Interactive communication between the vehicle and the central station present enormous difficulties to the operating system. These difficulties stem from the needed flexibility of the operating system to manage incoming requests for assistance and interactively communicate with the vehicle. An example is an operating system of a present day monitoring station. This station receives a first distress call from a first vehicle. The operating system assigns the call to a first available on duty operator positioned at a monitoring station, i.e., a computer with a monitor. A second distress call is received from a second vehicle and the operating system assigns that call to a second available operator at a second monitoring station. This process continues and the operating system will continue to assign calls to the first available operator without regard to the nature of the call or the proficiency of the selected operator. The operating system of this example merely monitors and assigns. It does not manage the available resources. This type of operating system randomly distributes operator and

monitor resources.

A random distribution of resources yields an operating system that is hardware dependent, i.e., computers, display monitors, and telecommunications equipment. Randomly distributing resources does not consider operator skill or experience. Randomly distributing resources in this manner is assigning incoming calls to operators on a per chance basis. A further complication of resource allocation is the communication from the central station to the vehicle.

Various networks and communication systems exist that can communicate with the vehicle. Each of these systems may be in a different associated format and has different protocol standards. Each manufacturer of these systems may have particularly required communication devices such as cellular phones. The manufacturer often wishes to maintain the distinction and uniqueness of its own product and standards to encourage the user of that particular device to remain within that structure. The operating system must accommodate those multiple standards. Moreover, as each technical advancement spurs additional changes in communications standards, it appears to be unlikely that a standard protocol and format of data transfer can or will be adopted. It is important to an operating system to be flexible and accommodate to new and different forms of technology. An operating system that randomly distributes resources can only accommodate change by increasing the number of distributed resources.

Modeling of a mobile security problem, as discussed above, is a far reaching task. The variables involved are numerous, interwoven, and complex. A technique that is understood by those skilled in the art of object-oriented analysis and design (OOA&D) will bring order to the seemingly unmanageable task of vehicle and personal security. The basic premise for OOA&D is set forth in *Object-*

Oriented Analysis & Design with Applications by Grady Booch, 1994. A brief overview of the Booch notation that is used in this disclosure is illustrated in Figure 9.

5 A class icon 91, FIG. 9, represents a view of the class structure of the system and is illustrated by an amorphous blob. A utility procedure icon 92, may represent a collection of free programs (non-member functions) that supply the interface to real world entities. A class category 93 represents a collection of related classes. Relationships between classes are represented by an
10 association line 94, inheritance arrow 95, "has" a solid circle line 96, and "uses" an open circle 97. Communication links between calling object 98 and called object 92 are illustrated by an order message arrow 99 and an object/value open circle 90.

15 Provided below are definitions of some object-oriented terms used in this disclosure:

- base class: the most abstract class from which all other classes are derived;
- super class: a class from which other classes are derived;
- 20 • subclass: a class that is derived from one or more super classes;
- persistence object: an object that can live longer than the program that created it;
- mutex: a software mechanism used in multi-threaded programming to synchronize the use of limited resources among all of the threads that
25 need access;
- ODBC: open data base connectivity;
- C: Class and object names have a prefix of "C" denoting what follows is the class or object function, example Cmodem.

A brief tutorial of the principles of OOA&D and the terms used herein will be given. The major principles of OOA&D will be explained in turn but in summary the principles are abstraction, encapsulation, inheritance.

5 Abstraction is the principle of ignoring those aspects of a subject that are not relevant to the current purpose in order to concentrate more fully on those that are relevant. Most of what is dealt with in the real world — people, places, things, and concepts are intrinsically complex. When abstractions are used it is understood that subject matter contains additional details but those details are
10 ignored in order to reduce the overall complexity of the subject matter.

 Encapsulation, also called information hiding, is a principle used when developing an overall program structure. Each component of the program structure should encapsulate or hide a single design decision. The interface to
15 each module is defined in such a way as to reveal as little as possible about its inner workings.

 Inheritance is a mechanism for expressing similarity among classes. Inheritance portrays generalization and specialization, making common attributes
20 and services explicit within a class hierarchy. Inheritance allows an object instance to benefit from hidden data or method in one or more superior objects. An object that inherits data method for more than one object is said to have multiple inheritance.

25 An object model is a collection of principles that form the foundation of object-oriented design. There are four major elements that form the object model: abstraction, encapsulation, modularity, and hierarchy. There are three minor elements of the object model: typing, concurrency, and persistence. All of these

elements, major and minor, constitute the framework of the object model.

Once the object model is developed, numerous software languages may be used to implement a chosen solution to a problem domain defined in the performance of OOA&D. The actual semantics of a chosen programming language such as C++, Smalltalk, or Ada may all be used to implement the object model that will result from the OOA&D. An ordinary person skilled in the art of any one of the above listed programming languages will be able to develop a software system derived from an object model. That software system is constructed by creating a number of source code programs that are compiled and linked together to form executable modules.

Object-oriented software systems and object-oriented programming languages are being increasingly applied to the solution of large system software problems. Object-oriented approaches are directed to the subdivision of a problem to smaller, more easily managed pieces. Object-oriented languages and systems encourage programming and development in terms of "objects". Objects implement encapsulation by providing a single definition of the data and process, i.e., known methods necessary to manage that object.

An object is an abstraction of something in the domain of a problem or its implementation, reflecting the capabilities of a system to keep information about it, interact with it, or both. It is an encapsulation of attribute values and their exclusive services.

A class is a description of one or more objects, with a uniform set of attributes and services. In addition, a class may describe how to create new objects in the class.

Encapsulation of an object requires that object data be manipulated or viewed by the use of one of the object methods. Each object is provided a set of operations providing a visible interface to other objects. Communication between objects is accomplished by message passing where messages conforming to the visible interface specification are accepted and processed by that interface. A message passed to an object is analyzed by the object that itself selects a method by which it will react to that message.

Object-oriented concepts allow significant reusability of objects through the concept of instantiation. Instantiation allows a copy or instance of an object to be created from either a generic object description or another object using the principles of inheritance. The created object has its own data and is permitted to override particular data in the generic object definition.

Accordingly, a fully functional operating system for a mobile security system may include a central monitoring station element, a vehicle security element, and a telecommunications network that interactively connects both elements of the mobile security system. The central monitoring station has an operating system with an architecture to accommodate changes that may and will occur in the future. The operating system manages the distribution of resources within the central station.

Summary Of The Invention

The present invention is a mobile security system that provides personal and remote unit security. The preferred system contains at least one central site monitoring station and a universe of remote units. The remote units are connected to the central site monitoring station by a telecommunications medium. That medium is preferably a cellular technology or other technology that allows communications between an individual remote unit and the central site monitoring

station. The remote unit comprises a vehicle containing a portion of or all of a protection system that provides security for the vehicle and its occupants. The protection system comprises, in part, a plurality of interactively communicating application programs that independently perform dedicated tasks and collectively provides security for the remote unit and the unit's occupants.

The central site monitoring station comprises, in part, a plurality of computers, monitors, and telecommunication equipment, all interactively and network connected. The operation of the central site monitoring station is implemented by an object-oriented operating system that commands and controls all aspects of the central site monitoring station. The object-oriented, operating system of the present invention resides in at least one of the afore-mentioned computers. For a matter of convenience the object-oriented operating system of the present invention residing at central site will be referred to as the central site operating system (CSOS). The CSOS comprises, in part, a plurality of interactive and independent functioning object-oriented application programs that all share a database. As a matter of convenience certain objects are grouped into easily identifiable sections. These sections are a communication server, a resource manager, an administrator, and a monitor.

The communication server provides a plurality of objects that connect and communicate with the remote unit. When a remote unit has a reason to communicate with the central site it transmits a reason for the call message (reason-for-call) to the central site. That call is received by the communication server. The communication server has already established a communication link to facilitate the incoming reason-for-call. The communication server also provides objects to interactively communicate with the resource manager and other sections of the present invention.

The resource manager provides a plurality of object classes that control the interactivity between the communication server, the database, and the monitor.

The resource manager facilitates and manages operators assigned to a selected workstation, monitors calls in process, and provides the communication server with its own connectivity requirements to other sections. One such object class employed by the resource manager represents a list of operators. This operator list object class is implemented by the resource manager to manage operators that are assigned to a selected workstation containing the monitor.

The monitor provides a plurality of object classes that enable a visual representation of the reason-for-call. That visual representation may include the current status of the remote unit and its location. The monitor also manages the interactive communications and responses to the remote unit. The monitor has a temporary interdependence on the resource manager and the communication server. This interdependence begins with the communication server requesting the resource manager to assign a selected operator and workstation to a given reason-for-call placed by the remote unit. Once this assignment is made a connection from the communication server to the workstation is complete. The resource manager then takes an inactive role until another request is received.

The monitor then seizes control of the operational functions of the workstation. Any programs in operation on the monitor are forced to the background. The monitor launches a plurality of object classes that allow the operator assigned to the workstation to communicate with the remote unit.

Once the monitor receives the reason-for-call, it displays an interpreted version of that call on a display of the workstation viewed by the selected operator. The monitor also displays all pertinent information concerning the remote unit.

The monitor allows interactive communication between the operator and the remote unit's occupant. The communication may be voice and/or data transfer. The communication may also be silent, i.e., portions of the remote unit may be in a listen mode.

5

The communication server, upon first activation by the operating system, establishes a data link layer by instantiating a CModem and a CPort object class. The CPort object class represents a selected port on the communication server. The CModem class represents the modem attached to that selected port on the communication server. When the communication server receives the reason-for-call from the remote unit the communication server creates a CVirtualRemoteUnit object. The CVirtualRemoteUnit object transforms the communications or transmittals from a selected communication device operationally disposed within the remote unit into transmittals recognizable by the CSOS. The CVirtualRemoteUnit object also translates commands and controls understandable by the operating system into commands and controls understandable by the remote unit.

10

15

20

25

The CVirtualRemoteUnit object represents the remote unit as if the remote unit was physically attached to a communication server. A CvirtualRemoteUnit object adjusts its behavior according to the product and version of the remote unit that placed the reason-for-call to the CSOS. The CVirtualRemoteUnit object can be viewed as a virtual translator that allows communications between the remote unit and the CSOS. The communication device in the remote unit sends data and receives commands transparently to the communication server. Conversely, the communication server sends commands and receives data transparently to the remote unit.

The monitor provides an interpretation or visual representation of the remote unit's status to the selected operator. This status contains remote unit location and information on the subscriber of the remote unit. The monitor also manages the reason for the call and invokes the appropriate response to the reason-for-call. This response may be the launching of the CRemoteUnitWindow object class. This particular class encapsulates the functionality needed to communicate commands to and receive messages from the remote unit.

The resource manager controls the interactions between the communication server, the database, and the monitor. One interaction may be facilitating the operator workstation that services the reason-for-call. The resource manager maintains an operator list by utilizing the COperatorList object. This class represents a list of operators and may be used by the resource manager to hold and manage a list of operators currently signed onto the system. Once the priority of the reason-for-call is established, an appropriate operator may be assigned to the particular reason-for-call.

The resource manager manages the interactive communication between the communication server and the monitor. The communication server has received a call from the remote unit and makes all the appropriate connections. The communication server calls upon the resource manager to facilitate the connection to a workstation so the reason-for-call may be processed. The resource manager invokes the COperatorList object to decide which operator is best suited to receive the incoming reason-for-call. The resource manager then assigns the appropriate operator to receive the incoming reason-for-call from the remote unit. The resource manager then takes an inactive position in relation to managing the interactivity between the monitor and the communication server. The resource manager also maintains a list of communication servers and ports

along with their present status and other information pertinent to their functionality.

The object-oriented applications of the present invention all share a (non object-oriented) relational database. The relational database employed by the present invention is a repository for data generated from operational activities of various application programs. The database also provides, on demand, a plurality of persistent data objects to various applications for the execution of tasks required by the CSOS.

The CSOS cannot access the database directly because an object-oriented application program cannot have static knowledge that existed before the execution of that particular program. If a presently operating application program desires to use a selected object that was stored by some other application program, it does not know the type of object because the operating program did not create that particular object. The present invention provides an interface between all of its application programs and the relational database used by the CSOS.

The interface module provides a persistent base class that has at least one method common to the persistent data objects stored in the database. A database object (persistent version of a system class) is derived from the persistent base class. The database object then overrides those methods to provide a new persistent system object class. The database object instantiates a query object that has at least one object containing a descriptor. The descriptor contains a plurality of information relevant to the stored data. The persistent system class has at least one database object; therefore, the persistent system class provides the object-oriented operating system with persistent data objects.

The database of the present invention provides a connection class, CDBConnection. This class provides a layer of abstraction between the interface module and the stored persistent data objects. Internal to the database this class uses an ODBC class library to perform all of its connection handling. This class is responsible only for keeping track of a valid connection handle to the database.

The benefits of using a database interface for access to a database from an object-oriented program have been generalized and encapsulated into the persistent base class and query classes. The programmer adding a new database object class to the system need only provide the descriptors for the queries. The queries are then generated at runtime of the particular application program.

Other features and advantages of the present invention will become apparent upon reading the following detailed description of embodiments of the invention, when taken in conjunction with the accompanying drawings and the appended claims.

Brief Description Of The Drawings

The invention is illustrated in the drawings in which like reference characters designate the same or similar parts throughout the figures of which:

FIG. 1 is a top level block diagram of the present invention;

5 FIG. 1a is a diagram of the environment in accordance with the preferred embodiment of the present invention;

FIG. 2 is a block diagram of the remote unit system in communication with a central site operating system;

FIG. 3 is a functional block diagram of FIG. 2;

10 FIG. 4 is a functional block diagram of the remote unit in communication with the central site system;

FIG. 5 is an object-oriented architecture of the central site operating system;

FIG. 5a is an object model of a communication server;

FIG. 5b is an object model of a resource manager;

15 FIG. 5c is an object model of a monitor;

FIG. 5d is an object model of an event processor;

FIG. 5e is an object model of a database interface module;

FIG. 6 is a block diagram of the typical components of the remote unit system;

FIG. 7 is a block diagram of the typical components of the central site system;

20 FIG. 8 is an illustration of an operator log on screen; and

FIG. 9 is an illustration of Booch notation.

Detailed Description Of The Preferred Embodiment

The interdependence of the element numbers in the drawing has been referenced above, and for the convenience of the reader will be reiterated here by citing an example of the flow of element numbers to drawings. This example is intended for illustrative purposes only: central site system 300, FIG. 1 is further illustrated at 300, FIG. 1a. The central site system 300, FIG. 1a includes a central site operating system 300a, FIG. 1a. The central site operating system 300a, FIG. 2 includes a communication server 300a20, FIG. 2. Communication server 300a20, FIG. 3 includes a port 300a20b, Fig. 3 ... etc.

A Mobile Security System

A top level view of the present invention is generally illustrated at 10, FIG. 1. The present invention 10, for the purposes of illustration only, is subdivided into 3 distinct groups: a central site system 300, a universe of remote unit systems 100, and a telecommunications system 200 interactively connecting subgroups 100 and 300.

The universe of remote unit systems 100 may contain at least one remote unit or may contain a plurality of individual remote units. The present invention 10 is not limited to the actual number of individual remote units comprising the universe of remote unit systems 100. The universe of remote unit systems 100 may be of different models or versions that may be produced by different manufacturing processes. The present invention 10 is not dependent on a homogeneous universe of remote unit systems 100. For the purposes of discussion in this disclosure only, the universe of remote unit systems 100 will be discussed as an individual remote unit 100 because the present invention will accommodate all versions of remote units. The individual remote unit 100 may be a vehicle of some type that contains one or more occupants. An example of a vehicle would be an ordinary passenger vehicle with or without a driver or any

occupant. The remote unit 100 may contain a portion of or all of a protection system that provides security for the vehicle, the driver, and its occupants. The protection system comprising, in part, a plurality of interactively communicating programs that independently perform dedicated tasks and collectively provides
5 security for the remote unit and the unit's occupants.

The central site system 300, FIG. 1, commands and controls all aspects of a central site monitoring station that is in communication with the remote unit 100. The central site monitoring station comprises, in part, a plurality of computers,
10 monitors, and telecommunication equipment, all interactively connected. A more detailed discussion of the central site system 300 will be provided.

Telecommunication system 200, FIG. 1, links together the central site system 300 and remote unit 100. The telecommunication system may contain at
15 least one medium through which voice and/or data may be transmitted. An example of that telecommunications link would be cellular phone communication or any other communication means that would allow at least one remote unit 100 to communicate with central site system 300. A more detailed discussion of telecommunication system 200 interactively linking central site system 300 and
20 remote unit 100 will be provided.

Environment Of The Mobile Security System

Communication from the remote unit 100, FIG. 1a, are received by the central site system 300 by communication network 300b. Communication network
25 300b may be any digital or analog technology that provides voice and/or data transmission to and from the remote unit 100 and the central site system 300. Communication network 300b is interactively connected to at least one of the central site computers 300d. Central site computers 300d may comprise a

plurality of computers. Central site computers 300d all have memories capable of storing programs. The actual number of computers employed in support of the central site system 300 is dependent on the overall desired size of the central site system 300. The present invention may be implemented on a single computer.

5 For the purposes of this discussion, central site computer 300d is discussed as if the present invention comprised only one computer 300d. Computer 300d integrates communication network 300b with at least one database 300c and at least one work station 300e. This integration of central site system 300 provides the framework for a central site operating system 300a to command and control all
10 aspects of mobile and personal security.

The central site operating system (CSOS) 300a may be modeled and derived by object-oriented techniques albeit the CSOS 300a is not dependent on object-oriented programming to function. In fact the CSOS 300a may be designed
15 and developed by any convenient or well-known programming method. The CSOS 300a controls the execution of programs that may provide services such as resource allocation, scheduling, input/output control, and data management. The CSOS 300a may reside on at least one computer albeit CSOS 300a may reside on a plurality of computers depending on the desired breadth of the central site
20 system 300.

Remote Unit System & Central Site Operating System

Operatively installed within remote unit 100 is a remote unit operating system 100a, FIG. 2. The remote operating system 100a comprises a remote unit
25 controller 100a20 that provides the structure to assimilate the data received from a plurality of sensors operatively functioning within the remote unit 100. The remote unit controller 100a20 receives data input from a stimulus from at least one sensor and responds accordingly. That stimulus may be from global position system

(GPS) buffer 100a40, a bidirectional communications and monitoring means 100a60 such as an audio response, or from the activation or deactivation of a communicator 100a80. An example of the stimulus would be a zone violation, i.e., an area of security within the remote unit 100 that is actively being monitored and that zone was entered or violated contrary to an established criteria. That criteria may be determined by the individual owner or subscriber of the remote unit 100, FIG. 1 or may be determined by the central site system 300 in conjunction with the subscriber. The remote unit controller 100a20 responds to the stimuli by formulating a reason-for-call message that is transmitted to the central site 300.

The central site operating system 300a, FIG. 2, has a communication server 300a20 that receives and transmits to and from remote unit 100. The communication server 300a20 requests a resource manager 300a10 to select an appropriate workstation 300e, FIG. 1a, that has an activated monitor 300a40, FIG. 2. The monitor 300a40 seizes control of the communication between the remote unit 100, FIG. 1 and the central site 300, FIG. 1. The operator of workstation 300e, FIG. 1a in conjunction with the monitor 300a40, FIG. 2, ascertains the appropriate solution or action that is required by the reason-for-call. This action may be retrieving a predetermined directive from the database server 300a60. That directive provides a solution for the reason-for-call from the remote unit 100, FIG. 1. An alternate means of providing a solution to the reason-for-call would be the operator to communicate directly with the remote unit. A communicator 300a80, FIG. 2 may be used by the operator of workstation 300e to communicate with an occupant or occupants of remote unit 100, FIG. 1.

Central Site Operating System

The communication server 300a20, FIG. 3 when first activated creates a port 300a20b that is assigned to a particular modem 300a20a. Communication server

300a20 may create a plurality of ports depending on the desired requirements of the central site system 300. The actual mechanism of the creation of port 300a20b to 300a20b -n is discussed hence forth. Modem 300a20a is selected from a modem pool comprising a plurality of modems. This modem pool is illustrated by
5 modem 300a20a to 300a20a -n, FIG. 3. The selected modem 300a20a may be dedicated to the transmission of voice, data, or a combination of voice and data. The actual selection of the mode of communications between the central site 300 and the remote unit 100 is dependent on the requirements of a particular transmission, i.e., the reason-for-call from the remote unit 100 and the central site
10 300. An example of data communication would be a desire by the subscriber of remote unit 100 to change the password that allows entry and control of the remote unit operating system 100a. The CSOS 300a acknowledging this request places the modem 300a20a in a receiving data mode allowing data transmission between the remote unit 100 and the central site 300. An example of voice
15 communications would be a desire by the subscriber of remote unit 100 to orally communicate with an operator of workstation 300e. The CSOS 300a, acknowledging this request, places the modem 300a20a in a receiving voice mode allowing voice transmission between the remote unit system 100 and the operator of workstation 300e.

20

Generally, the universe of remote units 100 may comprise a plurality of functioning systems. These systems may be the result of different types or upgrades of remote units that comprise the universe. The communication server 300a20 accommodates the variation that exists in the universe of remote units 100
25 by creating and then assigning a virtual remote unit 300a20c to a particular remote unit 100 that has requested communications with central site 300. The virtual remote unit 300a20c may be viewed or characterized as a translator of remote unit 100 language into a language the central site system 300 has currently

implemented. The actual mechanism for creating the virtual remote unit 300a20c will be discussed forthwith.

5 The CSOS 300a, FIG. 3, has an administrator program 300a70 that may be stored on one or more of a plurality of workstations 300e. That particular workstation would become an administrator workstation rather than a workstation dedicated to servicing remote unit system 100. The administrator program 300a70 is responsible for entering, viewing, and updating tables stored on database server 300c. The administrator program 300a70 supports generating
10 reports, billing for services, and providing supervising and monitoring of workstation operators.

A resource manager 300a10, FIG. 3, controls the interaction between the communication server 300a20, the database server 300c, and the monitor
15 300a40. One of the primary tasks of the resource manager is assigning an incoming call from the remote unit 100a to a selected workstation that has an appropriate logged-on operator. The resource manager 300a10 maintains a list of workstations, operators, and current activity for each workstation. The resource manager 300a10 also maintains a list of communication servers 300a20 and
20 ports 300a20b along with their present status and other pertinent information to their functionality.

The monitor 300a40 provides a visual representation of the remote unit 100 status to a selected operator. This status contains vehicle location and
25 information on the subscriber of the remote unit 100. The monitor 300a40 also manages the reason-for-call and invokes the appropriate response to the reason-for-call.

The monitor 300a40 activated on any selected workstation 300e may be engaged in other activities if the resource manager 300a10 has not assigned that particular workstation to service the reason-for-call generated by the remote unit operating system 100a. These activities may, for example, be an operator updating selected skills in conjunction with the administrator program 300a70. Another example would be the monitor 300a40 engaged in activities with any other programs in operation.

If the resource manager 300a10 receives a request from the communication server 300a20 to select an appropriate operator for a selected type of call, the resource manager 300a10 selects an appropriate workstation for that incoming call. The monitor 300a40 operating on that particular workstation will force other activities to the background, allowing the monitor to take control of the display. In a multi-tasking environment the activity is not halted but suspended until such time it is reactivated. The monitor 300a40 would then be enabled to communicate between the remote unit system 100 and the workstation 300e. The operator will now be able to view, on screen, information concerning the reason-for-call instead of any previous information.

The monitor 300a40 allows an operator of the workstation 300e to place a call out to remote unit 100. The operator places a call with the resource manager 300a10. The resource manager 300a10, transparent to the operator, contacts the communication server 300a20 and requests the next available time slot for communication with the remote unit 100. The resource manager 300a10 derives an estimated connection time from the communication server 300a20. This estimated time is transmitted to the operator.

The call out to the remote unit 100 is independent of the reason-for-call

placed by the remote unit system 100a. The operator may be actively involved in servicing the reason-for-call when the call-out feature is activated. When the communication server 300a20 places the call, the resource manager 300a10 is notified. The resource manager 300a10 contacts the correct monitor 300a40 and a pop-up screen is forced forward on the operators display indicating the connection is made and communication with the subscriber in the remote unit 100 is now possible.

An alternate method for calling out to the remote unit 100 by an operator of workstation 300e, if desired, would be the operator pinging the remote unit 100. This ping would activate the remote unit 100 and the remote unit itself would place a call to the central site 300. When the call is received by central site 300 the monitor 300a40 will contact the resource manager 300a10 to assign the incoming call to the appropriate operator. The resource manager 300a10 maintains a list of all operator generated call-outs to remote units. The resource manager 300a10 routes the incoming call to the operator that pinged that particular remote unit 100.

The monitor 300a40 may be actively engaged in responding to the remote unit 100's reason-for-call and be interrupted for some given reason. Regardless of the reason for interruption, the resource manager 300a10 suspends action on this call and maintains a suspense list with the caller's ID and other appropriate information. The monitor 300a40 minimizes the frame on the display screen of the associated operator. The remote unit 100 recognizes the transmission was interrupted and automatically attempts to reestablish contact with the central site 300. Upon successful contact with the central site 300 the resource manager 300a10, knowing this was an interrupted call from the same remote unit system, will assign this reason-for-call to the same workstation 300e. The monitor 300a40 activates the minimized suspended frame and the reason-for-call continues.

Remote Unit Operating System

The remote unit operating system 100a may be stored in the memory of controller 100a20, FIG. 2. The state and functional operation of the remote unit operating system 100a, FIG. 4 includes a plurality of interactive states that provide the intelligence for securing the remote unit 100. The interactive states also provide the functionality and capability to communicate with central site 300. The remote unit operating system 100a may contain a plurality of discrete prioritized functions that provide the remote unit 100 with the following features: security monitoring, position location, time stamping, remote unit status, low car battery, battery removed, carjack, carjack panic, remote panic, panic, call back, answer, check-in, GPS local, GPS limits, GPS fault, hand set fault, valet, valet limits, service, wrecker, request police, request EMS, information, key press, no hand set, request to call, power on self test, or remote unit configuration.

The remote unit operating system 100a is activated in the unarmed state 100a54. The unarmed state 100a54 performs no security functions for the protection of the remote unit 100 or its occupants. Transition from the unarmed state 100a54 to an armed state 100a58 may be active or passive. Active transition requires an identification of the subscriber of the remote unit 100. An example would be a specialized number or code that is unique to the subscriber such as personal identification number (PIN). Passive transition requires no subscriber intervention. The remote unit operating system 100a arms itself after a predetermined time period that is dependent on a selected mode 100a59. Mode 100a59 is determined by the subscriber at system installation or directly requested by the subscriber while user mode is activated. An example of this would be the remote unit 100 is in the unarmed state 100a54 and remote unit 100 is physically moved a distance predetermined by the subscriber. The remote unit operating system 100a, if in a carjack mode, will request verification of the subscriber's

identity. If this verification is not received by the remote unit operating system 100a in a timely fashion the unit becomes armed and takes appropriate action. The action may be sounding an alarm state 100a50, activating emergency procedures state 100a56, or requesting verification from an auditing state 100a52. The results of the audit state 100a52 may deactivate the response to the armed state 100a58 or proceed to contact the central site 300.

The remote unit operating system 100a may contact the central site 300. Each contact with the central site 300 comprises the transmission of remote unit status 100a44, combined remote unit status and GPS data 100a42, and data that is transparent to the subscriber. This data identifies the type of remote unit operating system 100a is presently activated on remote unit 100. The transmission medium may be any communicator 100a80 type that allows data and/or voice or a convenient combination of data and voice to be transmitted to the central site 300. The remote unit operating system 100a may be implemented by the aforementioned object-oriented techniques or by procedural techniques that are well known to those skilled in the art of programming.

Architecture Of The Central Site Operating System

An architecture of the CSOS is generally illustrated at 300a', FIG. 5. The communication server 300a'20 interactively communicates with the resource manager 300a'10 and the monitor 300a'40 through a remote procedure call layer (RPC) 300a'30. This remote procedure call layer 300a'30 provides a transaction process that facilitates distribution of resources, such as files and queues, throughout the system. CQueue 300a'32 and CInternalCom 300a'34 are typical objects that provide the RPC function that is well known in the art. The monitor 300a'40, the resource manager 300a'10, and the communication server 300a'20 are all connected to the database server 300a'60 through a relational object

interface module 300a'50. The interface module 300a'50 provides a layer of abstraction for the CSOS 300a'. This layer of abstraction facilitates database independence. The administrator program 300a'70 communicates directly with database 300a'60 and does not require interface module 300a'50.

Communication Server Object Model

The communication server object model 300a'20', FIG. 5a may comprise a plurality of object classes. These classes are designed for a flexible communication environment that may communicate with a plurality of different types of remote unit devices. An example would be objects that support ports 300a20b, FIG. 3, 300a20b -n up to 224 ports (224 ports are the maximum number of ports supported by a single WINDOWS NT server).

The CVirtualRemote class 300a'20'g, FIG. 5a, represents the personality of the remote unit 100 communicator 100a80, FIG. 2. The CVirtualRemote class 300a'20'g, FIG. 5a, provides a translation between communicator 100a80 codes and internal CSOS command codes and data structures. When the reason-for-call message is transmitted by the remote unit 100, FIG. 1, a CVirtualRemote object 300a'20'g is created. The CVirtualRemote object 300a'20'g is designed to facilitate that particular incoming call and is assigned to its own thread for multi-tasking purposes.

The principle responsibility of the CVirtualRemote object 300a'20'g, FIG. 5a is to recognize commands from the monitor 300a'40 and format those commands into messages that are recognizable by the remote unit operating system 100a. The same "turn siren on" command from the monitor 300a'40 is issued regardless of the type of remote unit 100 transmitting to the central site 300, and a command "turn siren on" from the monitor 300a'40 is by design a completely

different command from that which is sent to and recognized by the remote unit operating system 100a.

The communication server 300a'20', FIG. 5a process is facilitated by the
5 CommServer 300a'20'a object that directs the creation of new CVirtualRemote
300a'20'g objects. The CommServer 300a'20'a object allocates the
communications ports 300a20b to 300a20b -n, FIG. 3, through the use of Cport
300a'20'c objects. Each port, when allocated to a CPort 300a'20'c object, is
assigned its own process to service its input and output needs. All of the ports
10 300a20b to 300a20b -n facilitated by a single CommServer 300a'20'a are listed
and stored on a CPortList 300a'20'b object. A CUser 300a'20'i object is used by
the communication server 300a'20 to dial out to the remote unit 100. This object
facilitates voice communication between the operator workstation and the remote
unit 100.

15 Each port 300a'20'c instantiates two other objects that it uses to facilitate the
hardware and software needs of communications. A CModem 300a'20'e object is
created to facilitate the communications with the actual modem device 300a20a
and takes on the personality of a specific modem that is physically attached. A
20 CProtocol 300a'20'd object is also created to facilitate the specifics of the
communication protocol such as packet transfers, cyclic redundancy check
characters (CRC), calculations and error checking.

The CModem 200a'20'e objects are also kept on a CModemBank 300a'20'f
25 object that is a list of CModem 300a'20'e types. The CModemBank 300a'20'f class
allows for specific sets of modems to be referenced as a single unit.
CModemBank 300a'20'f allow for flexibility in assigning modems as input, output,
or input-output types.

The communication server 300a' also makes use of an internal communication 300a'20'h object that allows it to communicate directly with the rest of the CSOS 300a'.

5 Resource Manager Object Model

10 The resource manager object model 300a'10', FIG. 5b, is the "traffic cop" of the CSOS 300a. Its purpose is to make certain that any one workstation and port resource do not become overloaded with excessive reasons for call messages. To accomplish this task, the resource manager 300a'10' comprises a plurality of object classes.

15 The primary point of contact with the outside world is the principle object (or thread) called resource manager 300a'10'a. Through an internal communication object class 300a'10'b, FIG. 5b, the resource manager 300a'10'a receives requests for information about available resources from the communication server 300a'20.

20 The resource manager 300a'10'a keeps a number of static objects for information about the entire system and for access to the database server 300a'60. The database connection object 300a'61 is the primary interface to the database objects that resource manager 300a'10 acquires to process requests from other modules. The license information object 300a'10'c contains information about the licensing limits for the specific installation of the CSOS 300. This information allows the resource manager object 300a'10'a to limit the number
25 of workstations, ports, communication servers and operators that it allows the system to facilitate at one time. The system configuration object 300a'10'd provides additional information about the system setup. The reason-for-call object 300a'10'e provides a list of "reasons for call" that the resource manager 300a'10'a

uses to determine how to allocate operators/workstations to a given
CVirtualRemote 300a'20'g object.

5 The resource manager 300a'10'a keeps five list objects in memory that it
uses to maintain the current state of the entire CSOS 300a'. These five lists are
the communication server list, the port list, the work station list, the call list, and the
out call list.

The communication server list 300a'10'f object contains a list of
10 communication server 300a'10'g objects that are currently active in the CSOS
300a'. For each communication server object 300a'10'g, there is a port list
300a'10'h object that contains a list of port 300a'10'j objects that the given
communication server 300a'20 has allocated and opened. By traversing these
lists, the resource manager 300a'10 is able to report on the current state of all
15 communications equipment in the system, and allocate ports for outgoing calls to
requesting workstations 300e to 300e -n.

The resource manager 300a'10 also keeps a workstation list 300a'10'k
object that contains a set of workstation 300a'10'l objects. This is a list of all
20 workstations that have signed onto the CSOS 300a'. For each workstation object,
there is an incoming call list 300a'10'm and an outgoing call list 300a'10'n object.
These lists are maintained so that the resource manager 300a'10 can determine
how many calls a given workstation is processing at any given time, and so enable
the resource manager 300a'10'a to efficiently balance the load of incoming calls
25 among all of the available workstations.

Monitor Object Model

The monitor object model 300a'40', FIG. 5c, represents the primary user

interface to the CSOS 300a. From the monitor object model 300a'40', calls may be received or sent through the other system facilities. The monitor 300a'40 is made up of a plurality of object classes, some of which are specialized to the user interface, and others that are primarily informational objects acquired from the database interface layer 300a'48', FIG. 5d.

The VirtualRemoteUnitWindow 300a'40'a class, FIG. 5c, represents a class of graphical user interface counterparts that are instantiated by the communication server 300a'20'. The VirtualRemoteUnitWindow 300a'40'a provides the user interface elements to display information acquired from the remote unit 100 on a workstation.

The CVirtualRemoteUnitWindow object 300a'40'a instantiates the CEventQueue object 300a'48'b, FIG. 5d, through the event processor model 300a'48, FIG. 5. The CEventQueue object 300a'48'a provides an orderly method for events to be sent through the system. The event processor 300a'48 manages the internal connection between the monitor 300a'40 and the communication server 300a'20. The event processor 300a'48 facilitates all requests to the CVirtualRemoteUnit object at the communication server 300a'20 for information to be retrieved from the remote unit 100. Events coming from the CEventQueue object trigger these requests in an order specified by the directives in the database server 300a'60.

The monitor 300a'40 requires a single CDatabaseConnection object 300a'61 that allows all of the threads to access. The monitor 300a'40 controls access to the database connection through the use of a mutex to prevent more than one thread from accessing the database connection simultaneously.

The monitor 300a'40 establishes a CDirectiveMap object 300a'40'b when it is first activated. This map is acquired from the event processor 300a'48 and is discussed in that section in more detail. The directive map is used by all event queues that the monitor 300a'40 has instantiated as a means of quickly deriving a list of events for a given directive.

The monitor 300a'40 also creates an internal communication object 300a'40'b (in this case an RPC server) which is the port through which requests can be received from the communication server and resource manager 300a'10.

Other objects used by the monitor 300a'40 include, but are not limited by, the following database objects which contain information needed to handle incoming and outgoing call requests: subscribers 300a'40'e, user list 300a'40'h, vehicle 300a'40'f, and remote unit 300a'40'g.

Event Processor Object Model

The event processor 300a'48, FIG. 5 is further illustrated as an Event-Directive 300a'48' object model, FIG. 5d. This model is a simplified Booch diagram that exposes the underlying mechanism of the event processor 300a'48.

Events are discrete units of work that can be performed by the monitor 300a'40. Directives are aliases that represent groups of events placed in a predetermined order and stored on the database 300a'60. Event and directive object classes are not shown on the object model since they are the lowest level of data being utilized in this model.

The monitor 300a'40 when activated, instantiates a DirectiveCache 300a'48'a object and commands it to LoadCache (message 1). The

DirectiveCache 300a'48'a object responds by connecting to the database 300a'60 and querying the database 300a'60 for a list of all directives along with their associated list of events. Each directive is stored in the DirectiveCache 300a'48'a in an internal map structure so the associated list of events can be retrieved by a
5 call to FindDirective (message 3).

During runtime, the monitor 300a'40 receives information from the communication server 300a'20 that causes the monitor 300a'40 to send an ExecuteDirective (message 5) to an EventQueue 300a'48'b object. This message
10 causes the FindDirective (message 3) to load a set of events into the EventQueue 300a'48'b from the DirectiveCache 300a'48'a. Once the EventQueue 300a'48'b has been initialized in this way, the monitor 300a'40 begins periodically asking (message 2) the EventQueue 300a'48'b to DoNextEvent() which sends back to the monitor 300a'40 a pointer to the next event on the EventQueue 300a'48'b. If the top
15 event on the EventQueue 300a'48'b is a directive, then another call to FindDirective is made to retrieve the appropriate list of events.

Once the monitor 300a'40 has completed processing for any given event sent to it by the call to DoNextEvent(), it logs the resulting data produced by the
20 event to the database 300a'60. This is done through a call (message 4) to EventLog 300a'48'd. If the specified event is to be tracked for a period of time, this event information (m_sEventData) is stored in an event table on the database 300a'60.

25 When a call has been completed by the monitor 300a'40, it places an EndCall directive on the EventQueue 300a'48'b through another call to ExecuteDirective(). The EndCall directive contains an event that causes the EventQueue 300a'48'b to shutdown operations, and also causes the

RemoteUnitWindow 300a'48'c object to call ClearQueue() (message 6) to empty any remaining events that may be left in the queue. This final call makes certain that all events are properly deleted.

5 When the monitor 300a'40 shuts down totally, it sends ClearCache() (message 7) to DirectiveCache 300a'48'a to empty itself before deleting the DirectiveCache 300a'48'a object. This ensures all directive and event objects within the cache have been properly deleted.

10 Database Interface Object Model

 The CSOS 300a', FIG. 5a, cannot access the database server 300a'60 directly because an operating object-oriented program can not have static knowledge that existed before the execution of that particular program. If a presently running program desires to use objects that were stored by some other
15 program, the presently running program does not know the type of objects because the program did not create them.

 The principle difference between a relational data base system and an object-oriented programs is that an object "knows" what operation can be
20 performed on its data, whereas a relation database only has a set of generic operations that can be performed on its tuples. The semantics available in an object model are not preserved in a relational database. For example, in an object model, a "wheel" knows that it is related to an "axle" and inherits from a "wheel". In contrast, a relational database represents this information in three separate data
25 tables with no explicit representation of the relationships between the tables. The wheel table in a relational database might have foreign key information referring to the axle table, but this representation of the relationship between wheel and axle is implicit in an object model.

The CSOS 300a' provides an interface between all of its programs and the relational database 300a'60. Alternatively, an object-oriented relational data base may be used in place of the non-object-oriented relational database used by the present invention. The present invention is not limited to the type of database used to store information required by the CSOS 300a'.

The relational databases 300a'60, FIG 5, may store a plurality of object classes. An example a few of those objects are discussed below.

The CDBConnection 300a'61 class represents a connection layer to database server 300a'60. This class provides a layer of abstraction between the program and the actual database access method. Internally, this class uses an ODBC class library to perform all of its connection handling. This class is responsible only for keeping track of a valid connection handle to database server 300a'60. This object is public.

CSubscriber 300a'62 object represents a single subscriber to database server 300a'60. Embedded in this object are the methods used to store, retrieve, update, and delete subscribers within the database server 300a'60. Database server 300a'60 requires the presence of a CDBConnection 300a'61 object to perform its functions.

The CRemoteUnit 300a'63 object represents a single remote unit system from the database server 300a'60. Embedded in this object are the methods used to store, retrieve, update, and delete CRemoteUnit within the database server 300a'60. This object requires the presence of the CDBConnection 300a'61 object to perform its functions.

The CRemoteUnitVehicle 300a'64 object represents a single vehicle from the database 300a'60. Embedded in this object are the methods used to store, retrieve, update, and delete remote unit 100 within the database. This object requires the presence of a CDBConnection 300a'61 object to perform its functions. Examples of its attributes include VIN, make, model, year, and color.

The CDriver 300a'65 object represents a single subscriber or driver of remote unit 100 from the database 300a'60. Embedded in this object are the methods used to store, retrieve, update, and delete the subscriber of the unit within the database 300a'60. This object requires the presence of a CDBConnection 300a'61 object to perform its functions. Examples of attributes include ID, name, height, and weight.

The CLog 300a'66 class is used to write logs of various types. These logs may be written to database tables, flat files, or other locations according to the attributes of the CLog 300a'66 object. These objects can be instantiated and used by any other object as required.

The CEvent 300a'67 object represents a single event that can take place within the system. CEvent 300a'67 objects are instantiated by CEventlist 300a'47 objects. The CEventlist 300a'47 object is a utility object that stores predetermined events in a predetermined order. Examples of these events include request remote unit status, request GPS status, start emergency flashers, carjack, turn on siren, turn off siren, and talk/listen. CEvent 300a'67 is a top level class not to be used directly. Other event type classes are derived from it to provide functionality for specific event types.

The Interface Object Model 300a'50', FIG. 5e, illustrates the interaction of a

plurality of object classes that provide persistent versions of system or database 300a60 objects. These classes are all created to provide a generic and flexible architecture for adding new persistent classes to the CSOS 300a.

5 A database object 300a'50'a, FIG. 5e, a persistent version of a system class, is derived from a persistent base class 300a'50'd. The persistence base class 300a'50'd provides functions (or methods) that are common to all persistent object classes. The persistent base class 300a'50'd also provides methods to store and retrieve objects on the database server 300a'60. An example of these methods is
10 Retrieve() and Store(). Retrieve() and Store() is implemented in the Persistent base class 300a'50'd to facilitate the retrieving and storing of database server objects 300a'60. Each derived database object class 300a'50'a then override those methods to provide the details for a new persistent class. An example of the implementation of those methods would be a request for data concerning a
15 particular driver of a vehicle by a program running on the CSOS 300a'. The requested information resides in CDriver 300a'65 on the database 300a'60. The requesting program transmits a Retrieve() the database 300a'60. The request, transparent to the requesting program, is transmitted to the interface module 300a'50.

20 The mechanism that allows for actual database 300a'60 access is encapsulated in a query class 300a'50'c, FIG. 5e. The query class 300a'50'c generates a structured query language (SQL) statement based on a set of variables or elements (descriptor) that are used in processing the SQL statement.
25 The descriptor contains a plurality of information. Examples include a table name in the relational database 300a'60, column names to retrieve data types, and pointers to the locations where data is be stored. Each database object 300a'50'a has at least one query object 300a'50'c each containing a descriptor.

Data element object 300a'50'b, FIG. 5e declares each data element in the descriptor. The data element 300a'50'b object is a base class for which new data types can be derived. This allows flexibility in that any particular type of data element may be used within a query object 300a'50'c.

5

Once all the descriptors have been declared and the database 300a'50'a object instantiated, retrieving an object is a matter of sending a Retrieve() method. The Retrieve() method acts on each of the query 300a'50'c objects as required to obtain the needed information to build an object from the database 300a60. Each query 300a'50'c object uses its descriptor to build an open database 300a60 connection (ODBC) record set object. That record set object includes the SQL query required for transmission to the database 300a60 server. The database 300a60 server executes the query. Now the first result row is retrieved and the column values are transferred to the addresses pointed to by each data element 300a'50'b object. This process continues until all desired information is retrieved from the database 300a60.

10

15

The benefits of using the database 300a60 interface 300a'50 for access to the database 300a'60 from an object-oriented program have been generalized and encapsulated into the persistent base 300a'50'd class and query 300a'50'c classes. Thereby the programmer adding a new database 300a'50'a object class to the system need only provide the descriptors for the queries. The queries are then generated at runtime.

20

25 Typical Components Of The Remote Unit

Typical components that may comprise the remote unit 100aa, FIG. 6, are a controller 100a20a interactively connected a plurality of input and output devices 100a22a. An example of that controller is a Z-180 microprocessor. Any convenient

input or output device known in the art may be used to connect to controller 100a20a. These inputs and output devices 100a22a receive signals from zone sensors 100a23 and provide output signals to relays and other devices. Input and output devices 100a22a may be any convenient or common device known in the art. The transceiver 100a60a provides the communication link between remote unit 100 and the central site 300. An example of transceiver 100a60a may be Nokia 6080 cellular phone. To facilitate that communication link, transceiver 100a60a has a hand set 100a80a, a microphone/speaker 100a80a', and an antenna 100a80a". The location processor 100a42a provides remote unit 100 positional information to the present invention. An example of the location processor 100a42a is a Magellan AVI-10 GPS module. All of the components of the remote unit may be interactively connected by any convenient means that is known in the art.

15 Typical Components Of The Central Site System

An interconnecting schematic, FIG. 7, illustrates the typical components that may be used to implement the present invention. The typical components of FIG. 7 are all well known in the art and the interconnection of those components is of standard configuration known by those skilled in the art of network communications. Communication server 300d to 300d -n, workstation 300e10 to 300e10 -n, and Administrator Workstation 300e may be a plurality of Hewlett Packard Vectra 5/133 computers with display monitors. The database server may be an Oracle 7 database server operative on a Hewlett Packard Vectra 5/133 computer. The modems 300b32, 300b34, 300b36, and 300b38 may all be MultiTech V-34 Triple Modem external modems. Modem 300b31 is a digital modem that may be used to receive digital data transmitted by the remote unit 100. Those modems are interconnected to a modem manager 300b30, for example, a MultiTech MR 4800. The modem manager 300b30 is connected to a standard

PBX 300b20 system and a standard public switch 300b10. The present invention also contains network capabilities using a network hub 300b38 in conjunction with WINDOWS NT server software installed on one of the communication servers. Telephone 300b37 to 300b37 -n are standard and may be used to communicate with the remote unit 100 or with other desired locations. The central site 300 typical components illustrated in FIG. 7 and disclosed herein are for illustrative purpose only and do not represent all of the possible combinations of typical components available that effectively implement the present invention.

A preferred mode of operation of the present invention is to provide a mobile security system that has central site system 300 interactively communicating with remote unit 100 via telecommunications system 200. The remote unit 100 has at least one communication device 100a60a and an input device 100a22a. The input device 100a22a responds to a triggering stimulus such as a zone violation or intrusion into remote unit 100. Those input devices trigger or generate a reason-for-call message that is transmitted via the communication device 100a60a. The reason-for-call message has a predetermined priority level indicative of the nature of the triggering stimulus. The central site 300 has CSOS 300a operatively disposed in at least one operator workstation 300e with a display and memory. The central site 300 has at least one communication device for transmitting and receiving transmittals to and from the remote unit 100.

The CSOS 300a comprising, in part, the monitor 300a'40, the resource manager 300a'10, and the communication server 300a'20. The communication server 300a'20 contains a plurality of objects for connecting and communicating with the remote unit 100. The communication server 300a'20, once activated, communicates with the remote unit 100 by first instantiating CModem 300a'20e object by CPort 300a'20c object. This allows a connection between a modem,

such as 300b32, to a software port created by CPort 300a'20c object. The communication server 300a'20 derives from the reason-for-call, type and model number of remote unit 100. The communication server 300a'20 instantiates a virtual remote class 300a120g to establish a valid connection with the remote unit 100 via the software port. The communication server 300a120 requests the resource manager 300a'10 to assign an appropriate operator workstation that has an activated monitor 300a'40 ready to receive a reason-for-call message from the remote unit 100. The resource manager 300a'10 derives the appropriate operator based on a log-in event that occurs when the operator signs on to a particular workstation 300ea, FIG. 8. The resource manager 300a'10 then provides the communication server 300a20 with an address of a selected operator workstation. The communication server 300a20 will connect the selected operator workstation 300e to the remote unit 100.

The monitor 300a'40 then responds to the reason-for-call indicative of the nature of the triggering stimulus. This response may include activating the event processor 300a'48 to supply the monitor 300a' with a directive that corresponds to the triggering stimulus. This directive contains discrete units of work or events that can be performed by the monitor 300a'40. The monitor 300a'40 in conjunction with the operator take action in accordance with the supplied event to resolve the reason-for-call from the remote unit 100.

While the invention has been described in connection with certain preferred embodiments, it is not intended to limit the scope of the invention. The scope of the invention is intended to extend to alternatives, modifications, and equivalents as may be included within the spirit and scope of the invention as defined by the appended claims.

We claim:

1. A central site operating system for managing resources of a monitoring station which uses a telecommunication system to monitor a plurality of remote units, comprising:

- 5 a communication server program which manages communications between the monitoring station and said remote units and receives information transmitted by said remote units;
- a monitor program which manages the information sent from said remote units to a workstation;
- 10 a database server program which stores information about said remote units and monitors said resources; and
- a resource manager program which manages interactions between said communication server, said monitor program, and said database server.

15 2. A central site operating system as recited in claim 1, wherein said communication server program comprises a virtual remote program for communicating with said remote units.

3. A remote unit as recited in claim 2 comprising a one-way pager.

4. A remote unit as recited in claim 2 comprising a two-way pager.

5. A remote unit as recited in claim 2 comprising a cellular telephone.

20 6. A central site operating system as recited in claim 1, wherein said monitor program comprises an event processor for managing interactions with said communications server.

7. A central site operating system as recited in claim 1, wherein said operating system comprises objects, and said database server program comprises a

25 relational database and a relational/object interface for transferring data between said relational database and said objects.

8. A central site operating system as recited in claim 1, wherein said resource manager program comprises a communication server list, a port list, a workstation

list, a call list, and an out-call list, and wherein said resource manager program manages the resources of the operating system.

9. A remote unit operating system comprising:

a remote unit controller for receiving data from a plurality of remote unit sensors
5 and processing said data to derive a reason-for-call message to be transmitted to a monitoring station; and
a transceiver for sending and receiving messages to and from the monitoring station.

10. A process for a resource manager program to assign an in-coming call from a remote unit to a workstation, comprising the steps of:

maintaining a list of communication servers and ports and their on-line activities;
maintaining a list of operators, said workstations, and activities of said workstation activity;
receiving a call from said remote unit on said communication server;
15 parsing a reason-for-call from said call;
receiving requests from said communication server to connect to said workstation;
prioritizing said requests based on said reason-for-call;
matching said requests to one of said operators;
selecting said matched operator;
20 forcing a monitor application program to place said workstation activities in background mode to allow a monitor application program to display a reason-for-call from said remote unit; and
resuming pre-call workstation activity upon call termination.

25

1/15

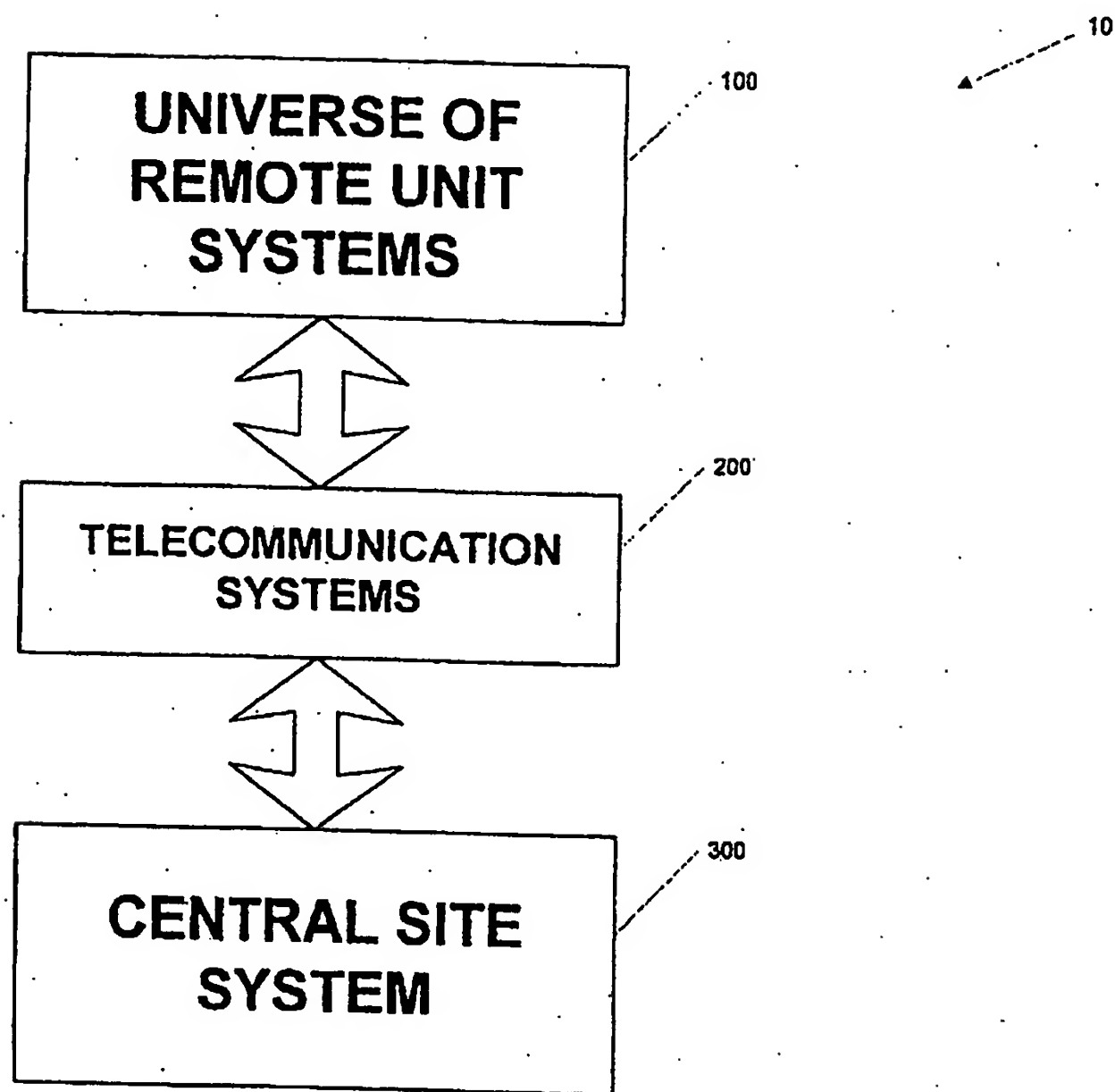


FIG. 1

SUBSTITUTE SHEET (RULE 26)

2/15

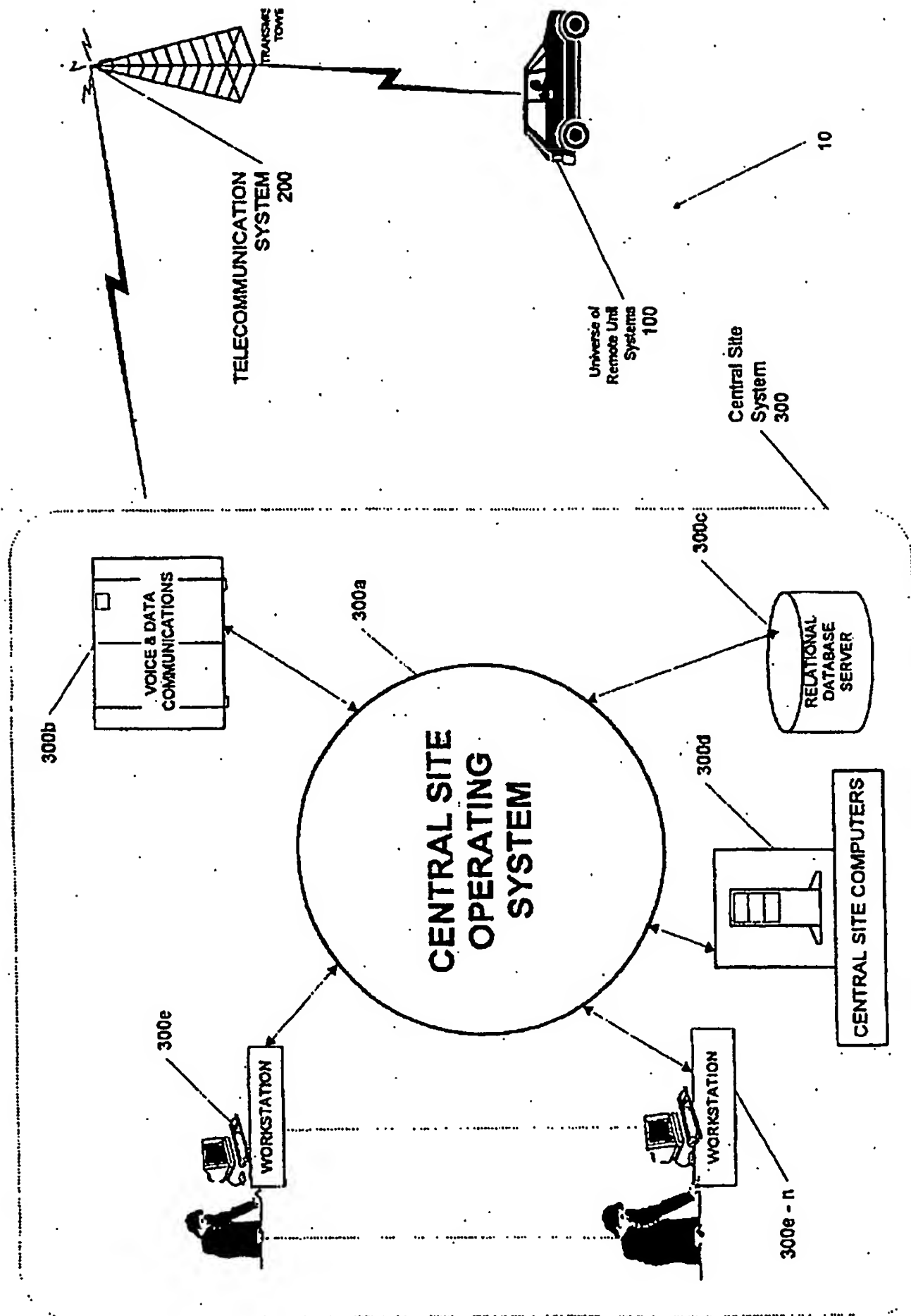


Fig. 1a

SUBSTITUTE SHEET (RULE 26)

3/15

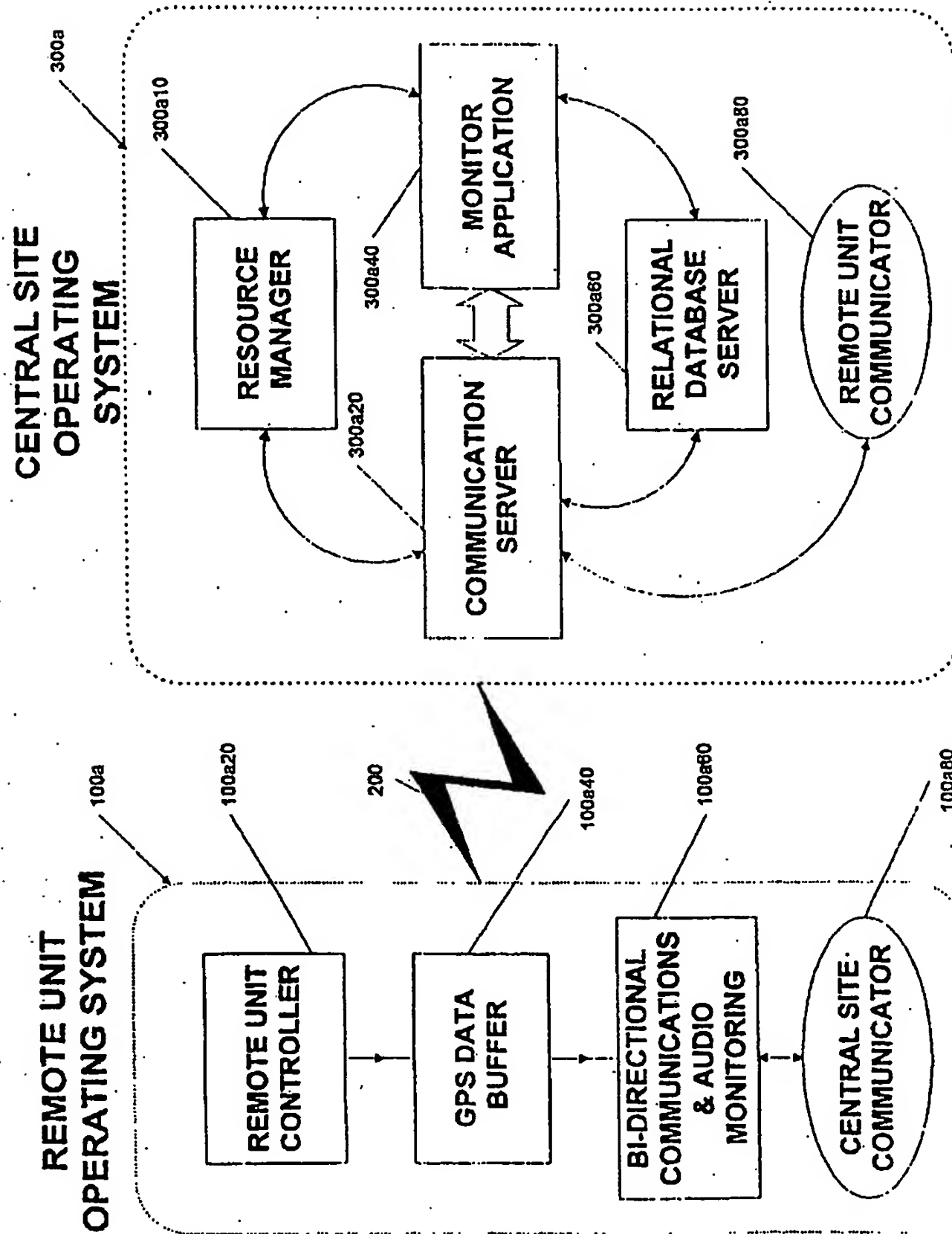


FIG. 2

SUBSTITUTE SHEET (RULE 26)

4/15

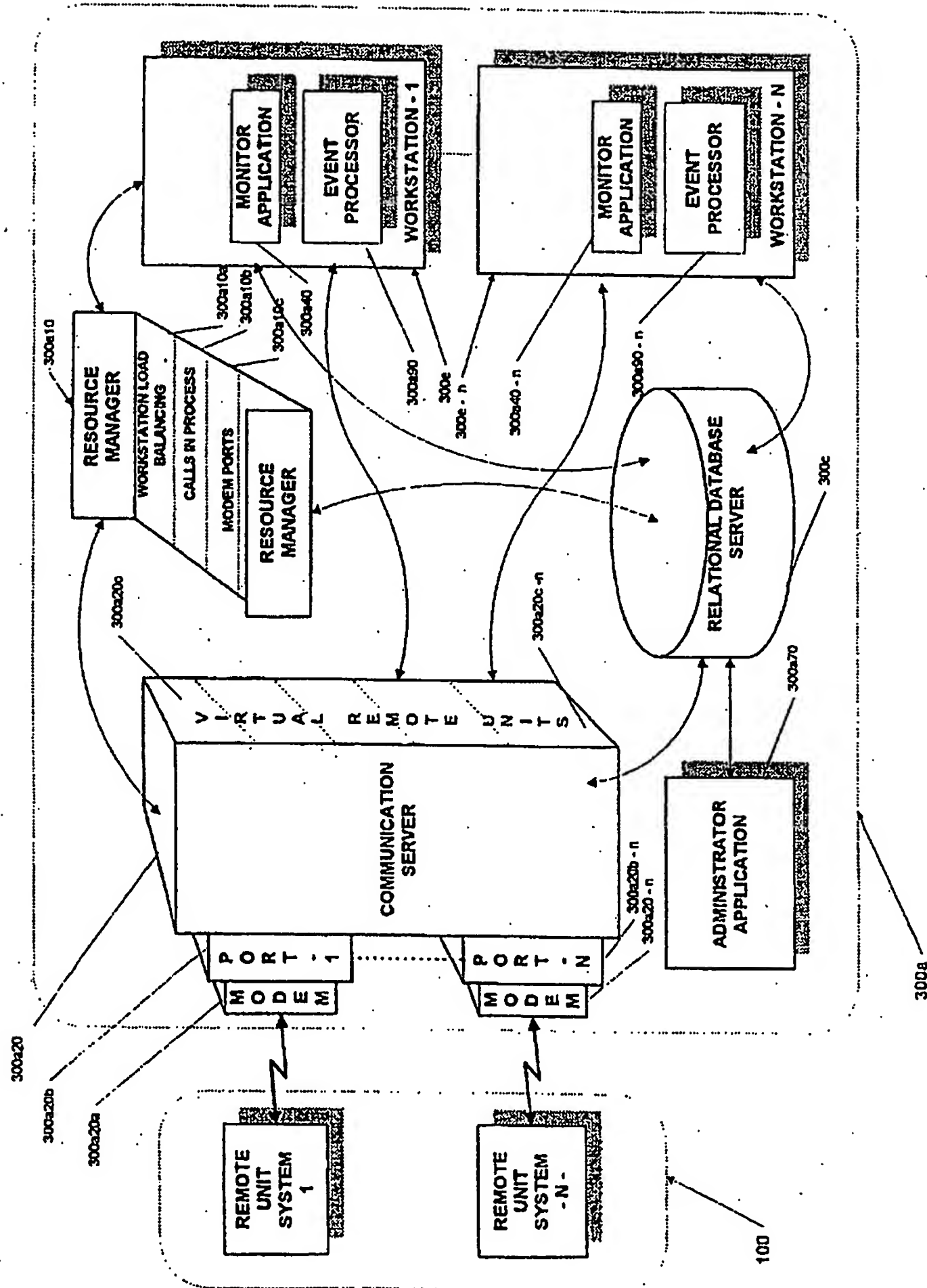


FIG. 3

SUBSTITUTE SHEET (RULE 26)

5/15

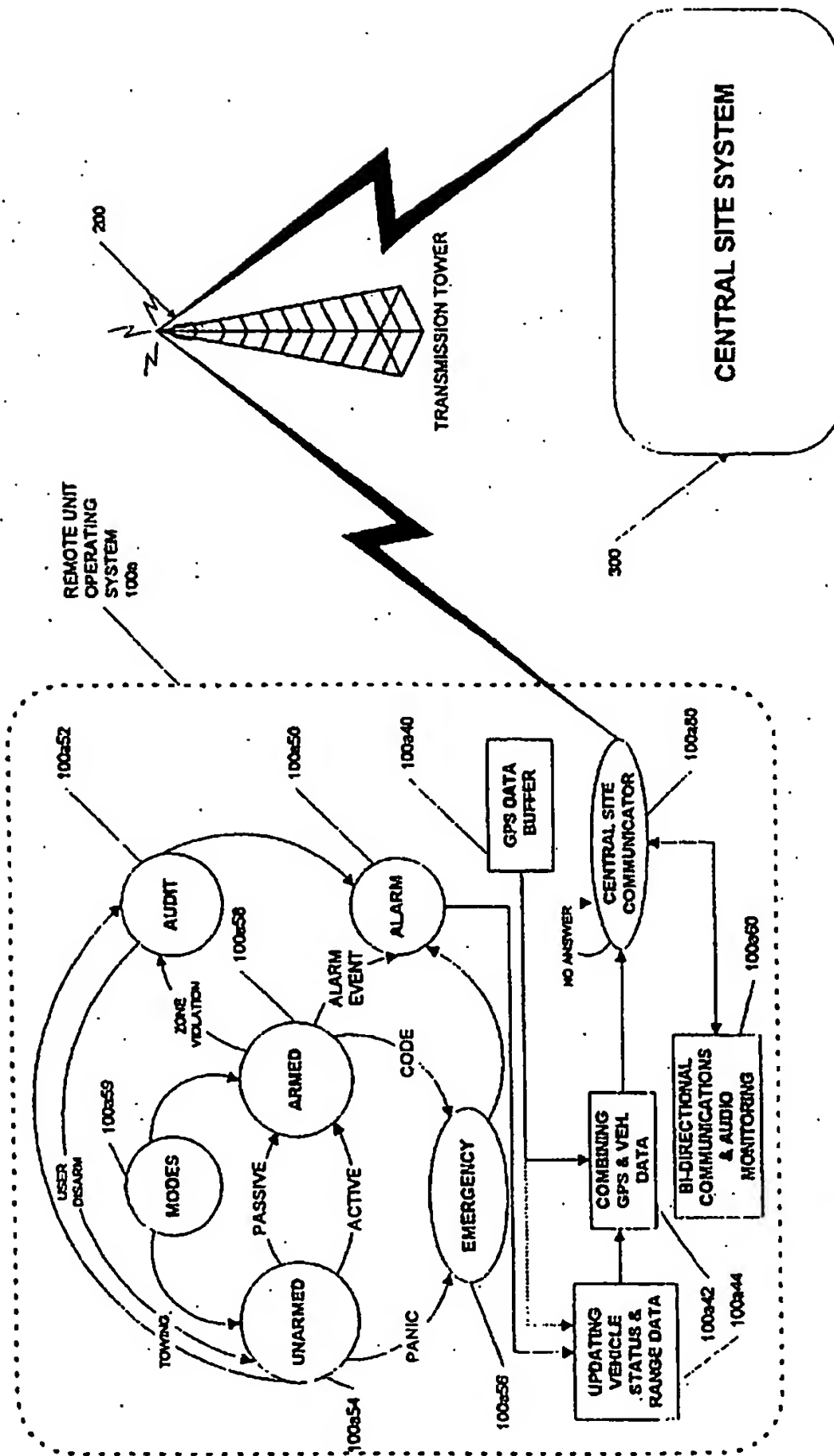


FIG. 4

SUBSTITUTE SHEET (RULE 26)

6/15

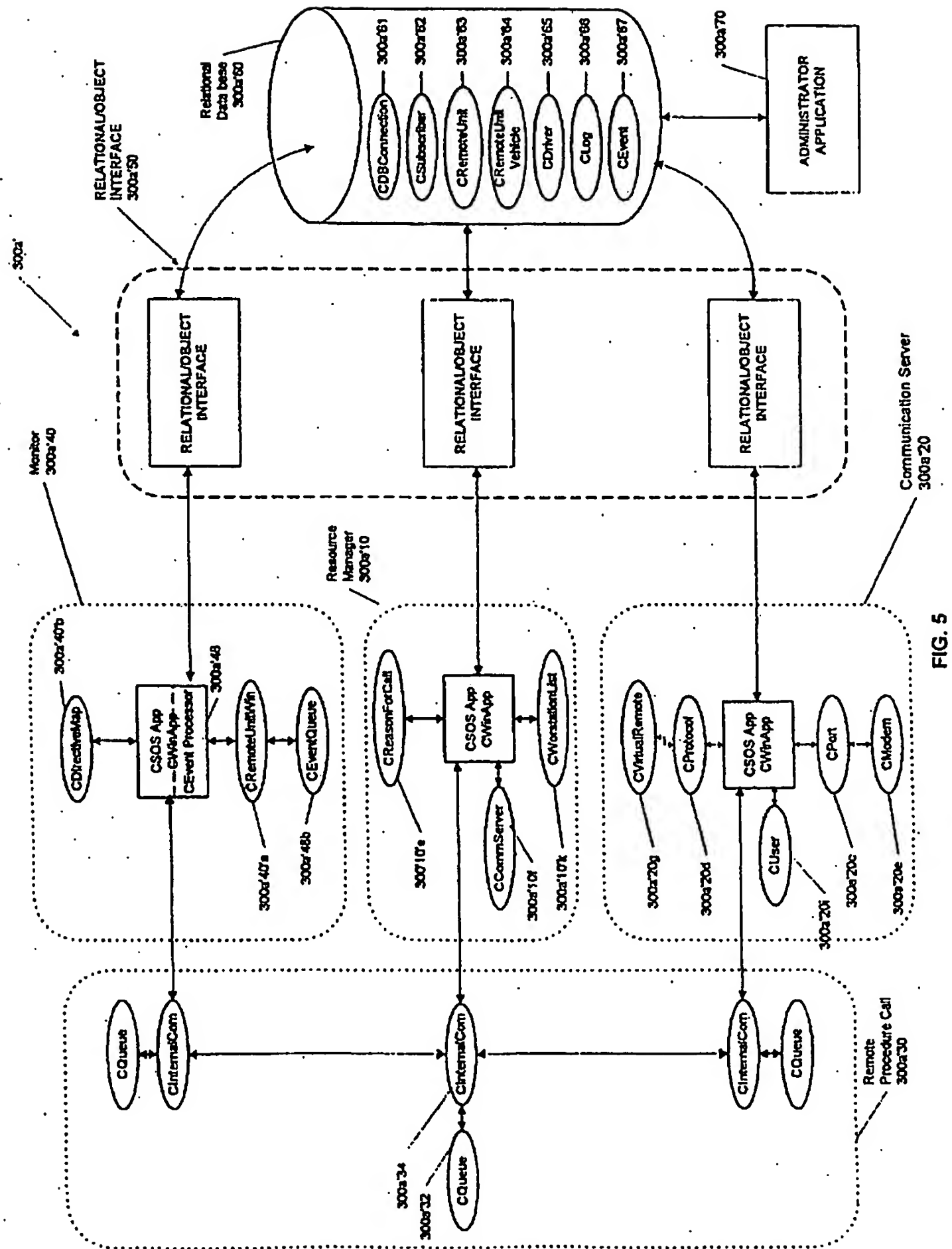


FIG. 5

SUBSTITUTE SHEET (RULE 26)

7/15

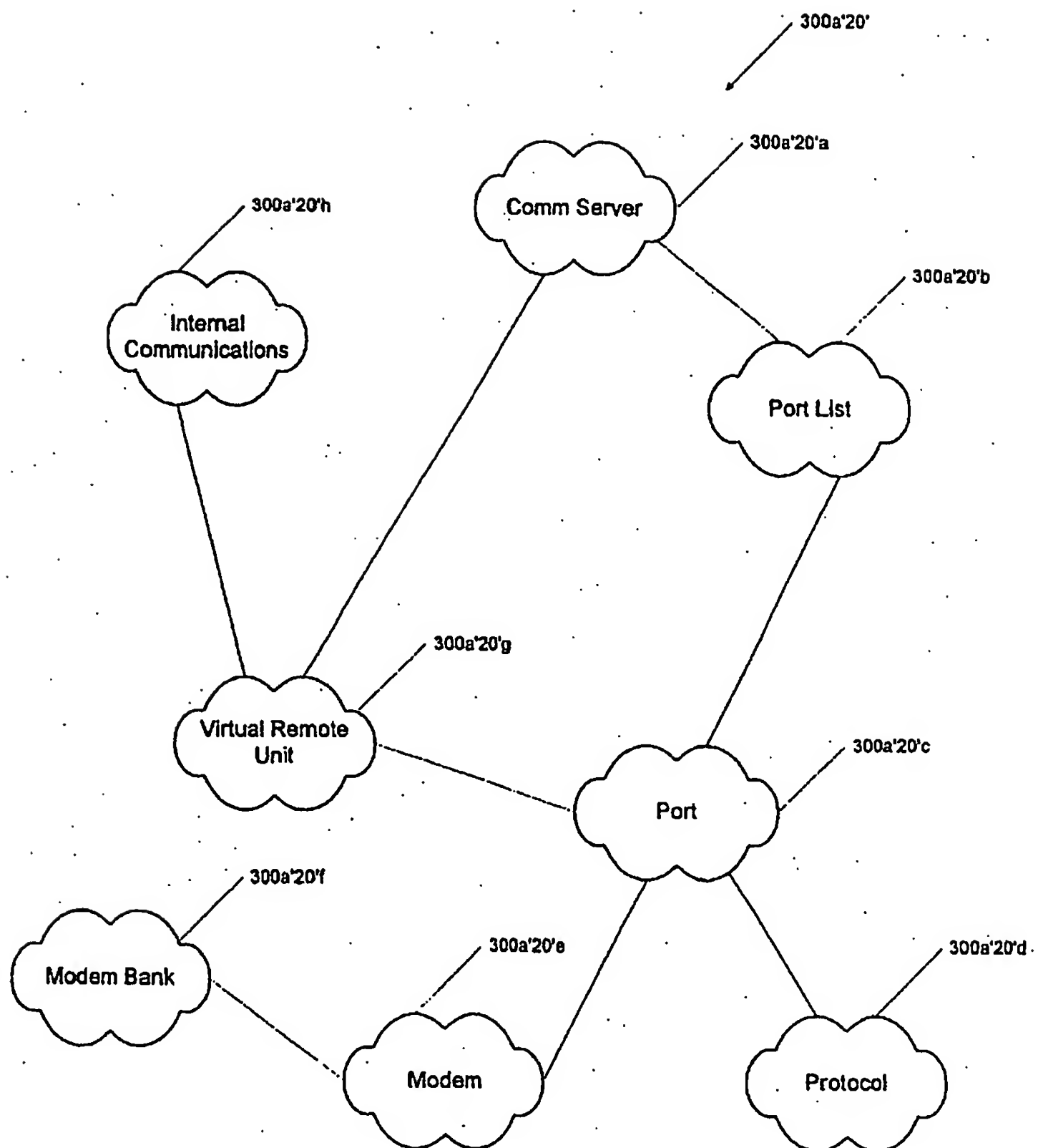


FIG. 5a

SUBSTITUTE SHEET (RULE 26)

8/15

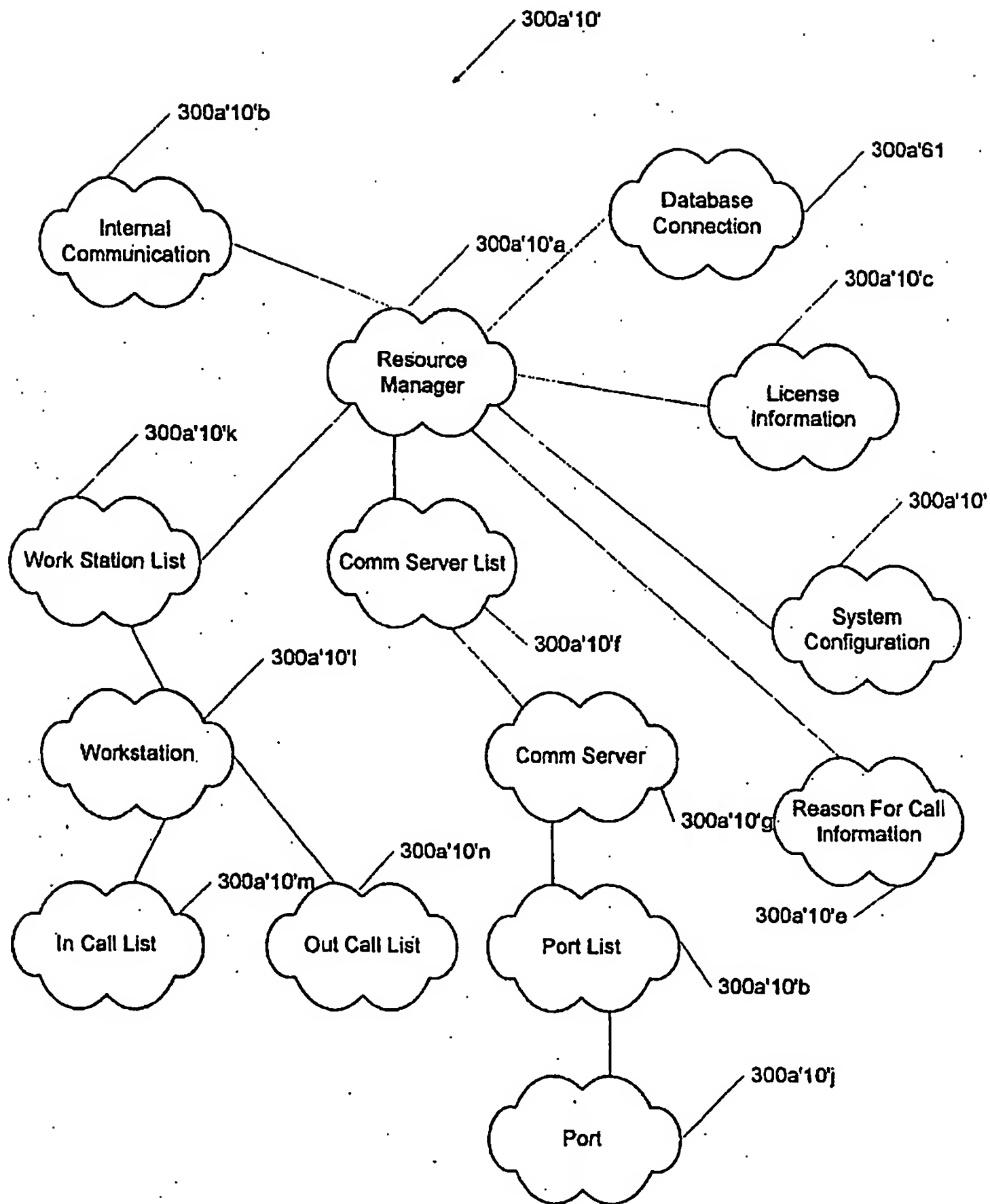


FIG. 5b
SUBSTITUTE SHEET (RULE 26)

9/15

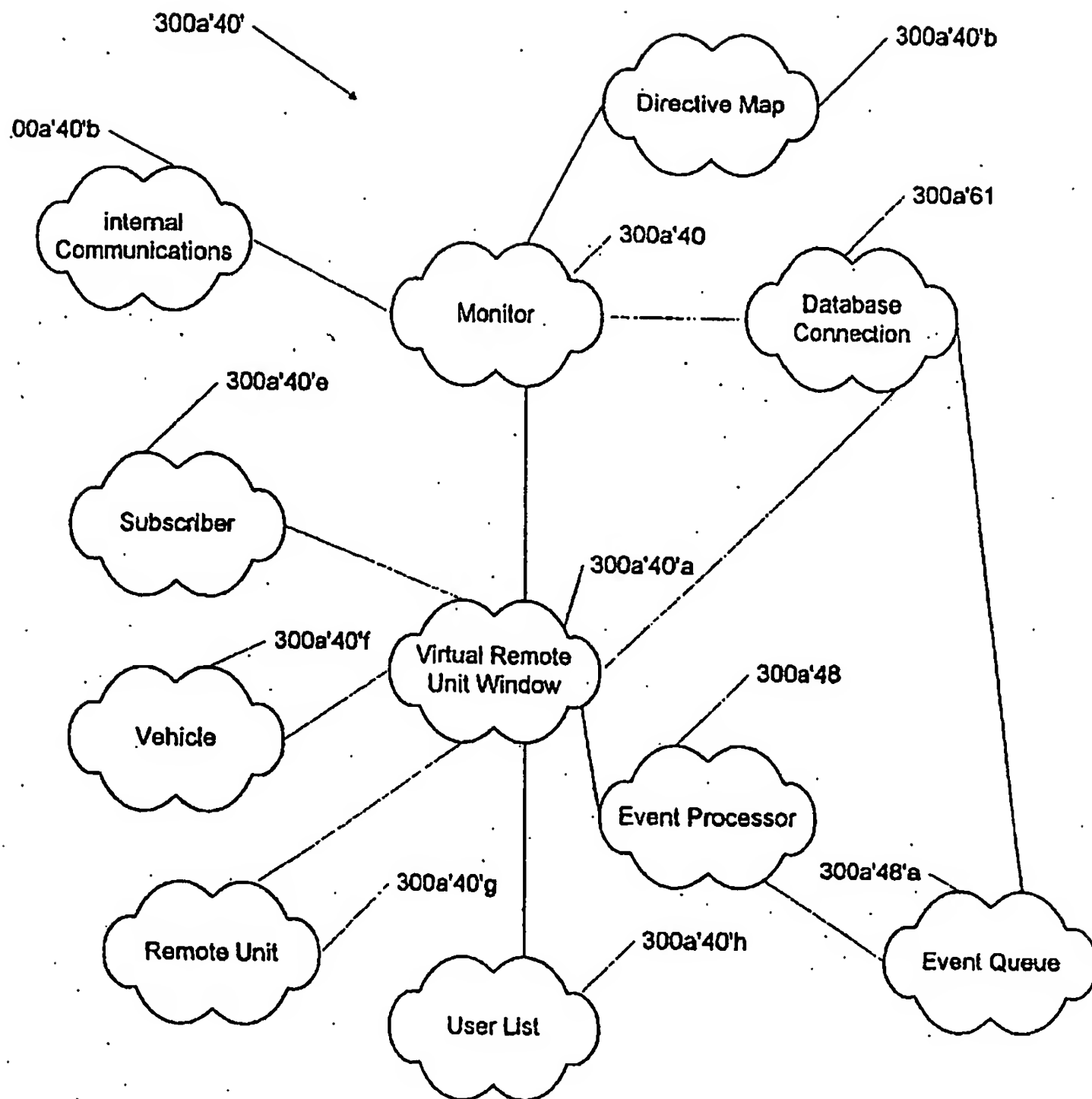


FIG. 5c
SUBSTITUTE SHEET (RULE 26)

10/15

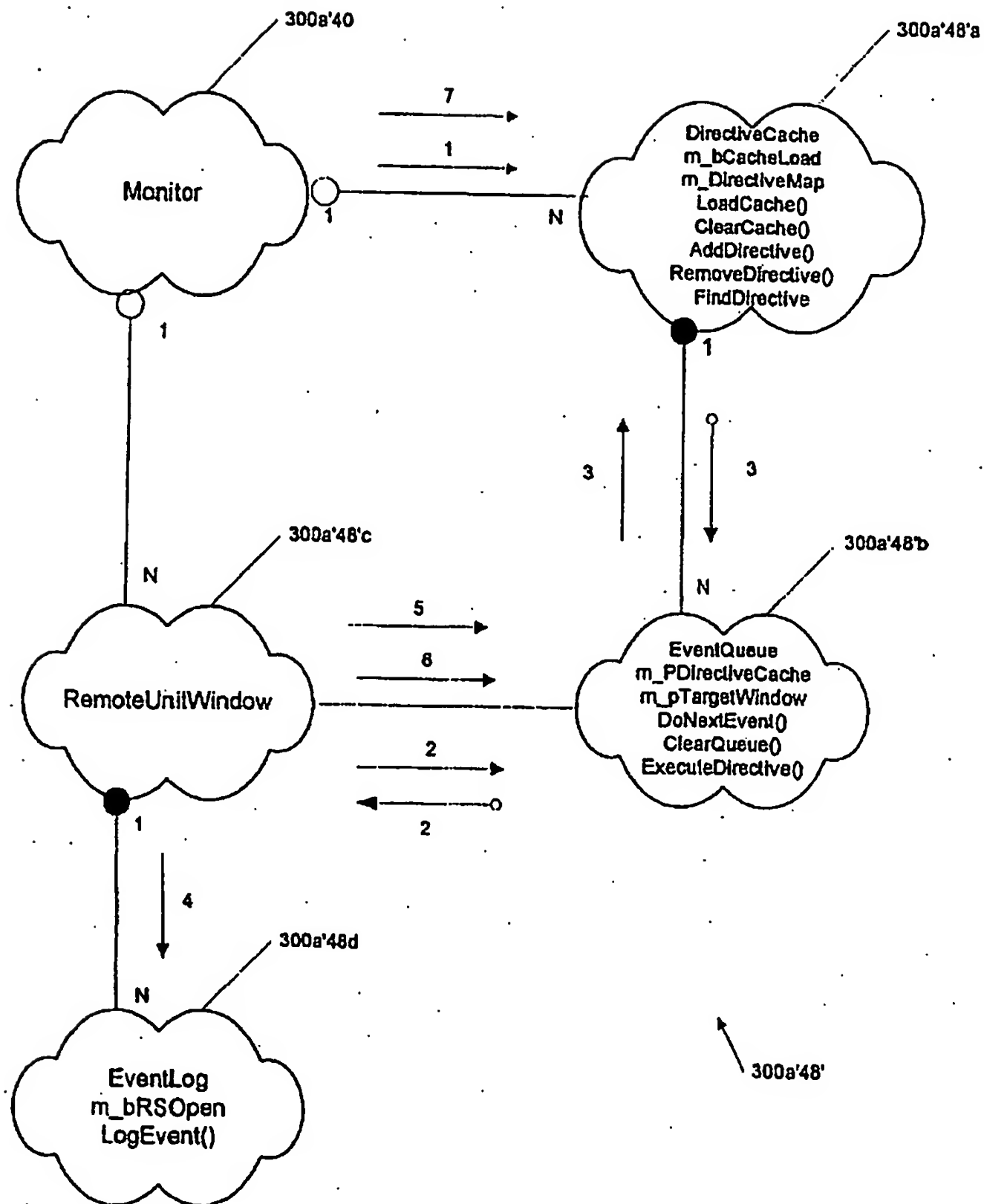


FIG. 5d
SUBSTITUTE SHEET (RULE 26)

11/15

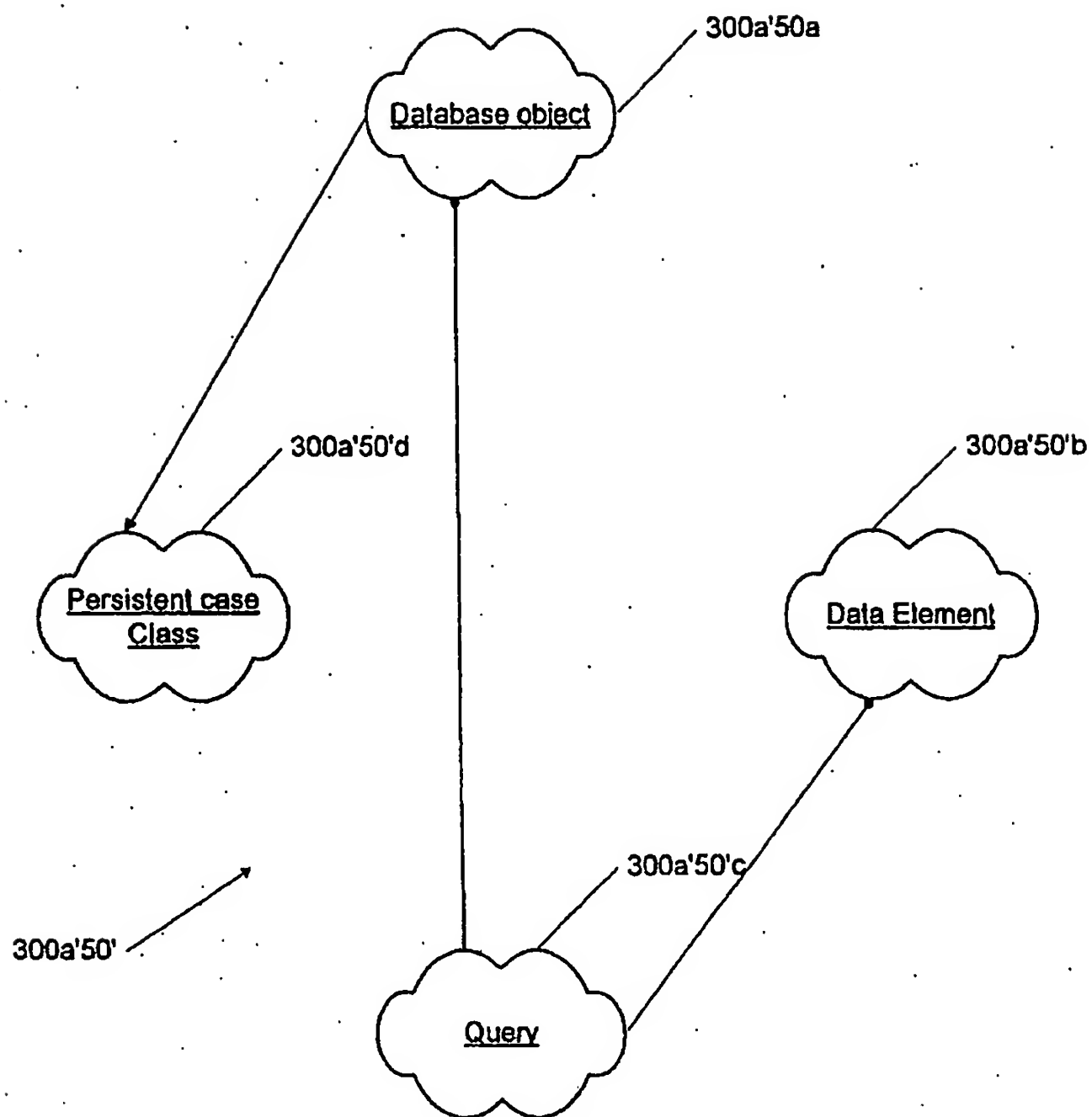


FIG. 5e

SUBSTITUTE SHEET (RULE 26)

12/15

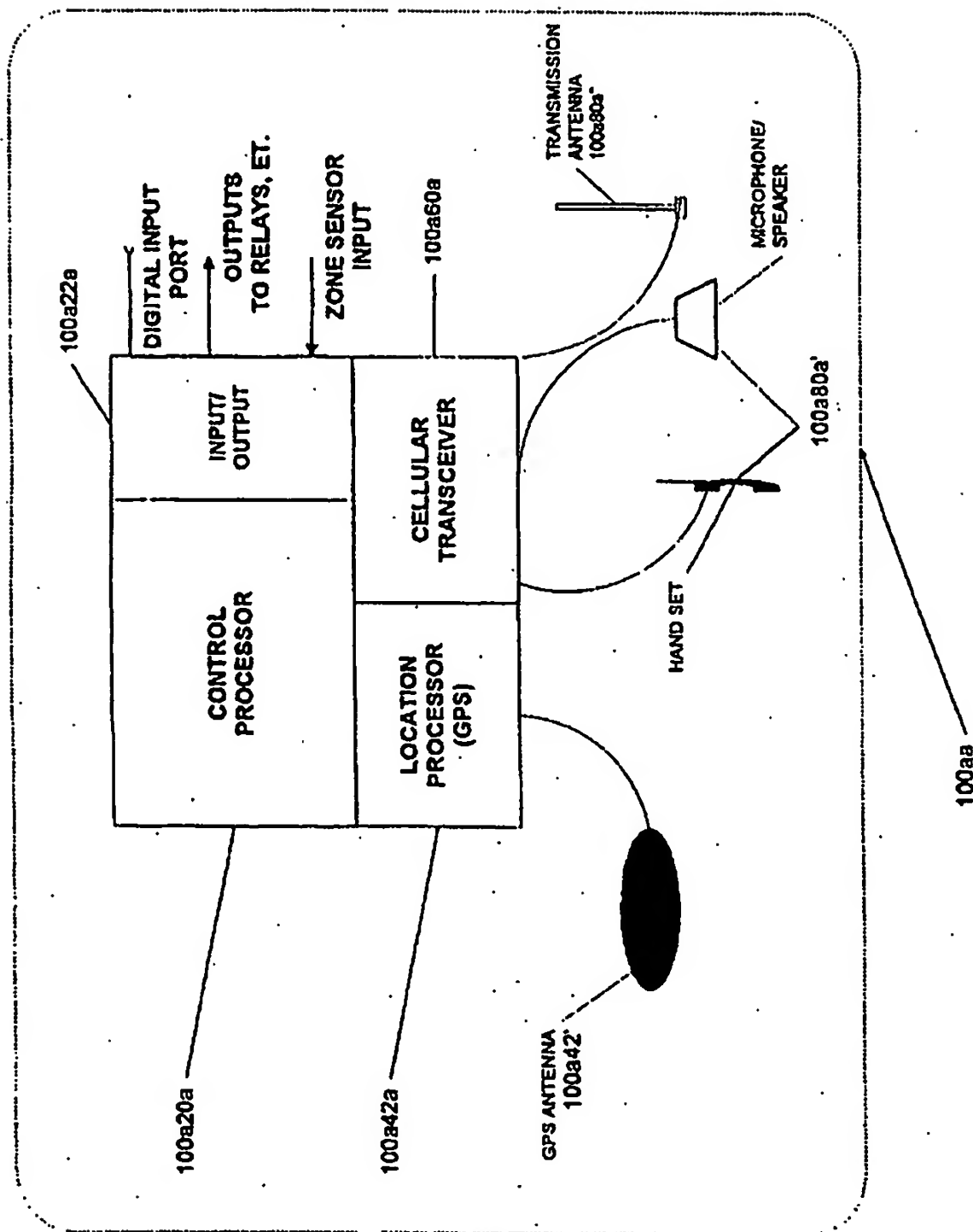


FIG. 6

SUBSTITUTE SHEET (RULE 26)

13/15

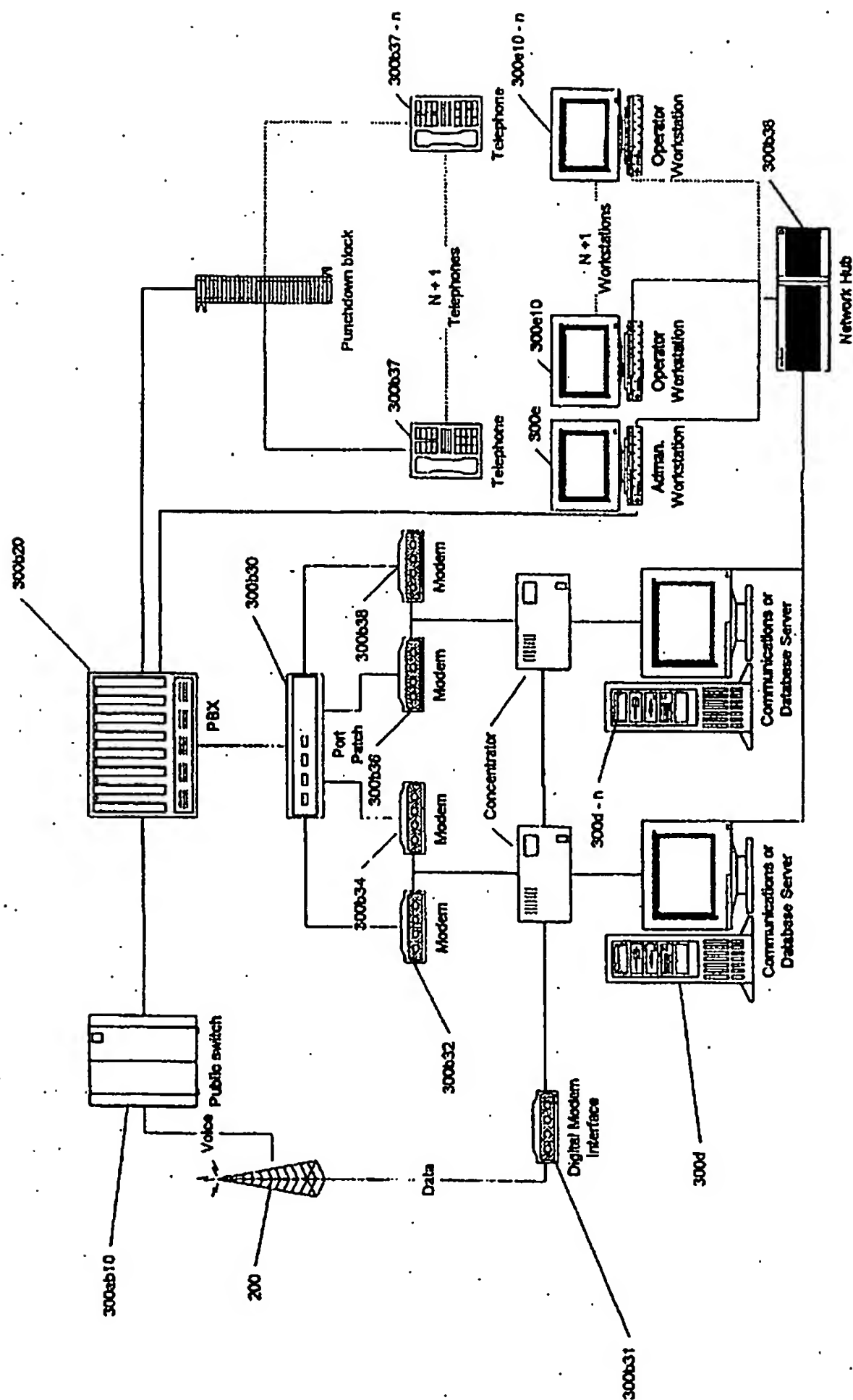


FIG. 7

SUBSTITUTE SHEET (RULE 26)

14/15

Workstation Operator Login

Operator ID

Password

OK

CANCEL

300ea

FIG. 8

SUBSTITUTE SHEET (RULE 26)

15/15

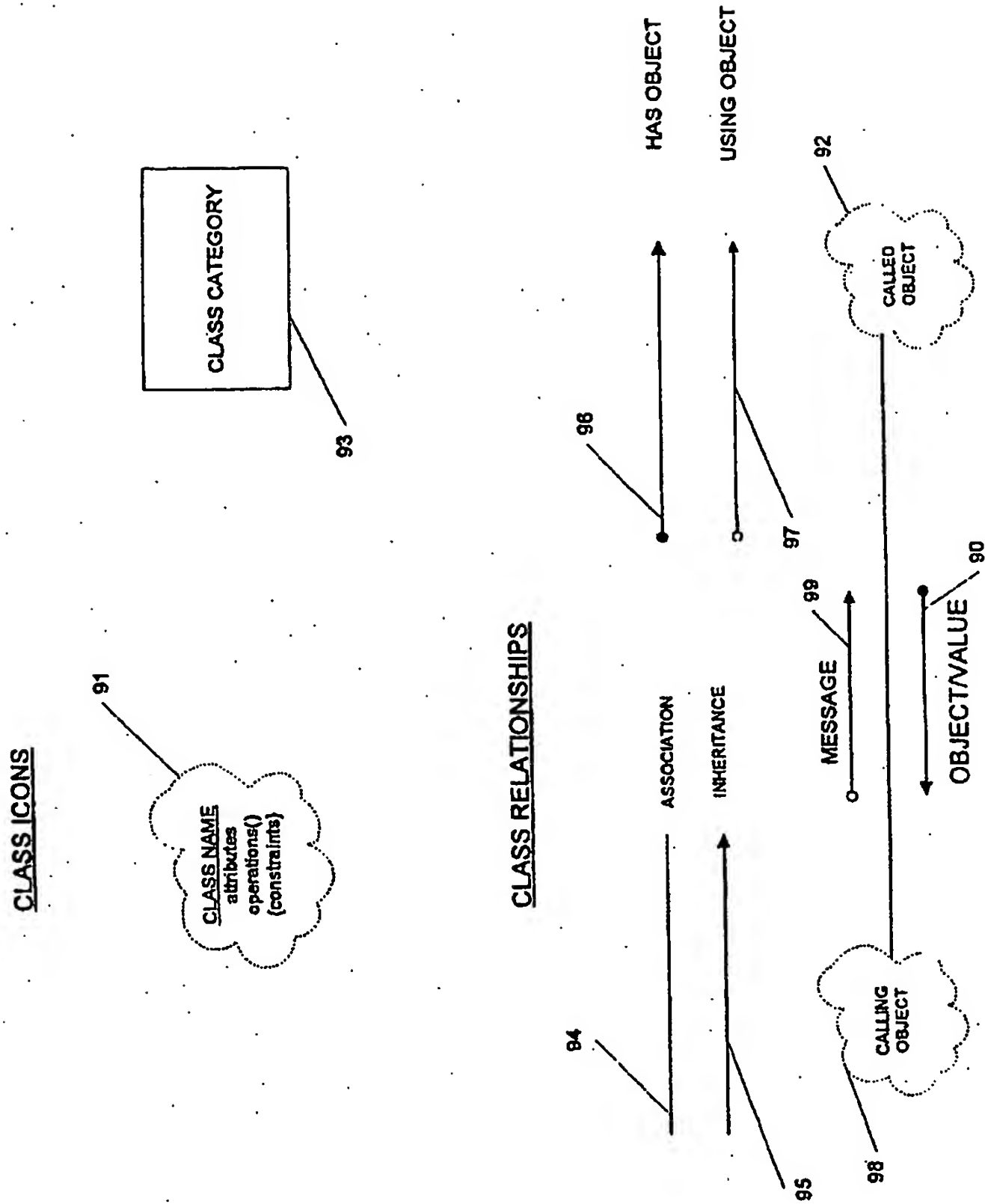


FIG. 9

SUBSTITUTE SHEET (RULE 26)

INTERNATIONAL SEARCH REPORT

International application No.
PCT/US97/15362**A. CLASSIFICATION OF SUBJECT MATTER**

IPC(6) : H04Q 7/00, 7/22, 7/20

US CL : 455/422; 379/45, 49, 39; 340/426, 425.5, 825.32

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)

U.S. : 455/422; 379/45, 49, 39; 340/426, 425.5, 825.32

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)

APS

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A,P	US 5,568,535 A (SHEFFER ET AL) 22 October 1996, col. 2, line 41 - col. 6, line 65.	1-10
Y,P	US 5,557,254 A (JOHNSON ET AL) 17 September 1996, col. 11, line 24 - Col. 14, line 18; col. 5, lines 24-30.	1-7,9,10
A	US 5,515,285 A (GARRETT, SR. ET AL) 07 May 1996, col. 14, lines 26-60, abstract.	1-10
Y	US 5,438,607 A (PRZYGODA, JR. ET AL) 01 August 1995, col. 5, line 18 - col. 6, line 37.	1-7,9,10

☐ Further documents are listed in the continuation of Box C. ☐ See patent family annex.

A	document defining the general state of the art which is not considered to be of particular relevance	*T*	later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention
B	earlier document published on or after the international filing date	*X*	document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone
L	document which may throw doubt on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)	*Y*	document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art
O	document referring to an oral disclosure, use, exhibition or other means	*A*	document member of the same patent family
P	document published prior to the international filing date but later than the priority date claimed		

Date of the actual completion of the international search 04 DECEMBER 1997	Date of mailing of the international search report 06 FEB 1998
Name and mailing address of the ISA/US Commissioner of Patents and Trademarks Box PCT Washington, D.C. 20231 Facsimile No. (703) 305-3230	Authorized officer DWAYNE BOST Telephone No. (703) 305-4778

Form PCT/ISA/210 (second sheet) (July 1992)*